Tripoli University

Faculty of Engineering

Computer Engineering Department

A graduation project is submitted in partial fulfilment of requirements for the degree of Bachelor in Computer Engineering

Design and Implementation of a Smart Parking System Based on Machine Learning License Plate Recognition on FPGA

By:

Sondus Giuma Edayakh

Supervised By:

Dr. Mohamed Muftah Eljhani

Fall 2022

Design and Implementation of a Smart Parking System Based on Machine Learning License Plate Recognition on FPGA

By: Sondus Giuma Edayakh

Approved by:

....

Dr. Mohamed Muftah Eljhani

....

Dr. Abdoulmenim Ahmed Bilh

....

Dr.

Abstract

The main objective of this project is to develop, simulate, and implement a car parking lot system for the College of Engineering at the University of Tripoli. The system designed using Verilog hardware description language (HDL) and deployed on the FPGA Cyclone IV GX EP4CGX150DF31C8 board. An essential feature of this system is its capability for automatic recognition of Libyan license plates.To monitor parking occupancy in real-time, the system utilizes an array of infrared (IR) sensors that are installed on each parking slot. These sensors provide continuous feedback on the availability of parking spaces. In addition, passive infrared sensors are employed to detect vacant slots. At the entrance of the parking lot, a digital camera captures images of vehicles. These images are then subjected to a series of processing steps using an automatic license plate recognition system. The system employs machine learning algorithms, specifically the aggregate channel feature (ACF) method, to identify the region containing the license plate and subsequently recognize the digits. To ensure accurate license plate recognition, the ACF detector was trained using a dataset consisting of 2,899 images of license plates. The system's performance was evaluated using 742 images of Libyan license plates, resulting in an average precision rate of 90%.

Acknowledgement

I am deeply grateful to Allah for granting us the knowledge, resilience, and capability to successfully finish this project. Without His mercy and guidance, I would not have been able to complete this task.

I would also like to express my gratitude to our families, friends, and colleagues for their unwavering support throughout this journey. I extend a special thank you to those who went above and beyond in providing extra assistance and encouragement.

I am indebted to my supervisor, Dr. Mohamed Muftah Eljhani, for his continuous guidance, support, and motivation throughout my project. I also want to thank everyone at the Department of Computer Engineering, particularly Eng. Yusra Maatug, for always being available to listen and encourage me.

Finally, I appreciate all those who assisted with this project in any capacity, and I apologize if I inadvertently omitted anyone. Thank you very much for your valuable assistance.

Contents

List of Figures

Acronyms

IR	Infrared
LED	Light Emmiting Diod
MATLAB	MAtrix LABoratory
ML	Machine Learning
ANN	Artificial Neural Network
FPGA	Field Programmable Gate Array
VHDL	Very High Speed Hardware Description Language
ACF	Aggregate Channel Feature
LPR	License Plate Recognition
HDL	Hardware Description Language
SVM	Support Vector Machine
OCR	Optical Character Recognition
RTL	Register Transfer Level

Chapter 1

Introduction

The College of Engineering at the University of Tripoli is currently facing a pressing issue concerning the surge in the number of vehicles within its parking premises. Consequently, effectively managing parking spaces and preventing unauthorized parking has become an arduous task [1].

The objective of this project is to devise and execute an intelligent parking system employing an FPGA (Field Programmable Gate Array). The system employs an array of IR (Infrared) sensors to promptly detect vehicle presence in parking slots and identify vacant spaces in real-time [2]. Additionally, the system utilizes a License Plate Recognition (LPR) mechanism that combines machine learning techniques and image processing methods. The LPR system analyzes captured images of vehicles entering the parking lot and deploys a MATLAB model employing Aggregate Channel Features (ACF) for license plate recognition and localization [3].

The proposed project for the smart parking system integrates a License Plate Recognition (LPR) system, which employs a fusion of machine learning and image processing techniques. The license plate recognition process encompasses four key steps: license plate localization, image pre-processing, license plate number extraction, and license plate number recognition [4].

To accomplish license plate localization, a machine learning technique is employed by the system. The image pre-processing step entails a combination of digital image processing techniques that prepare the license plate for number extraction while minimizing errors that may arise during the extraction process. The identification of license plate numbers is carried out through the utilization of an image classifier [5].

Compared to other license plate recognition systems, this particular system possesses a notable advantage by effectively accounting for varying illumination conditions, as well as different angles and distances from which the images are captured [6].

The project is divided into two main segments: the smart parking system and the Libyan license plate recognition system. The smart parking system aims to efficiently manage parking spaces by promptly detecting and responding to changes. On the other hand, the Libyan license plate recognition system concentrates on identifying and recording license plate numbers to enhance parking management and bolster security [7].

The structure of this report is divided into several chapters as follows:

Chapter 2: provides an explanation of the tools utilized throughout the project.

Chapter 3: review related works that are relevant to our project.

Chapter 4: will describe the methodology used for the first part of the project, which is the parking system. will also include simulation results as well as the outcome of the implementation on FPGA.

Chapter 5: we will detail the methodology used for the second part of the project, which is the license plate recognition system.

Chapter 6: will summarize the key points of our proposed system and discuss potential future enhancements that could be made.

Chapter 2

Background

2.1 IR sensor

It is an electronic device known as an infrared (IR) sensor is designed to detect and measure infrared radiation present in its surrounding environment. Infrared radiation, also referred to as infrared light, is a type of electromagnetic radiation that has longer wavelengths than visible light. It covers wavelengths ranging from approximately 1 millimeter to 700 nanometers, which is beyond the nominal red edge of the visible spectrum. Though it cannot be seen by the human eye, it can be felt as a warm sensation on the skin [8].

2.1.1 Types of IR sensors

There are two primary types of infrared sensors: active IR sensors and passive IR sensors. Active IR sensors work by emitting and detecting infrared radiation. They are composed of two parts: a light emitting diode (LED) and a receiver. The LED emits infrared light, which bounces off an object and is then detected by the receiver. As a result, active IR sensors act as proximity sensors and are frequently utilized in obstacle detection systems, such as in robotics.

In contrast, passive infrared sensors only detect infrared radiation and do not emit it from an LED. These sensors are made up of several components, including two strips of pyroelectric material, an infrared filter that blocks out all other wavelengths of light, a Fresnel lens that collects light from several angles into a single point, and a housing unit to safeguard the sensor from environmental variables such as temperature and humidity. PIR sensors are mainly used in PIR-based motion detectors [9].

2.1.2 IR sensors working principle

The operation of an IR sensor is similar to that of an object detection sensor. It is typically composed of an IR LED and an IR photodiode, which can be combined to form a photocopier or optocoupler. The IR LED emits IR radiation, which cannot be seen by the human eye, and is detected by infrared receivers in the form of photodiodes. The photodiode responds to the IR radiation generated by the IR LED, and the resistance of the photodiode changes in proportion to the amount of infrared light detected. However, it is important to note that IR photodiodes are different from regular photodiodes in that they only detect IR radiation. After the IR transmitter emits radiation, it reaches the object and some of the radiation bounces or reflects back towards the IR receiver. Based on the intensity of the response, the sensor output is determined by the IR receiver [9].

2.1.3 IR sensor Usages and benefits

Nowadays, IR sensors find applications in various fields, utilizing their diverse functions. For example, speed sensors are employed for synchronizing the speed of numerous motors, and temperature sensors are utilized for industrial temperature regulation. Passive infrared sensors, also called PIR sensors, are utilized in automatic door-opening systems, whereas ultrasonic sensors are utilized for measuring distances. In the College of Engineering/University of Tripoli Smart Car Parking System Based on Plate Recognition utilizes IR sensors to measure the empty vehicles lot. IR sensors provide several advantages, including low power consumption, a simple design, and user-friendly features. Additionally, IR signals are invisible to the human eye, making them an ideal choice for the parking system [9].

2. 2 Verilog Hardware Description Language

Verilog is a type of language known as a Hardware Description Language (HDL). It is utilized to describe digital systems, such as microprocessors, memory, flip-flops, and network switches. HDL allows for the description of any digital hardware at any level, and it is technology-independent. HDL designs are simple to design and debug and are often more beneficial than schematics, particularly for larger circuits.

Verilog supports a design at many levels of abstraction. The major three are:

- Behavioral level
- Register-transfer level
- Gate level [10].

2.3 MATALB

MATLAB, which stands for "MATrix LABoratory," is a programming language and computational environment created by MathWorks and owned by the company. It provides a range of capabilities for working with matrices, including manipulation, plotting of data and functions, algorithm implementation, creation of user interfaces, and integration with other programming languages [11].

While MATLAB is primarily designed for numerical computing, it also offers an optional toolbox that enables symbolic computing using the MuPAD symbolic engine. In addition, Simulink, an additional package, provides dynamic and embedded systems with model-based design and graphical multi-domain simulation capabilities.

2.3.1 MATLAB applications

MATLAB is centered around the matrix as its fundamental data element. The software provides toolboxes that are designed to help you achieve your goals by providing a collection of functions and/or classes. These toolboxes are created by professionals and are tailored to specific topics, enabling you to turn your ideas into reality.

- Statistics and machine learning (ML)

The toolbox in MATLAB is very useful. It allows for the easy implementation of statistical methods like descriptive or inferential methods, as well as machine learning. With this toolbox, various models can be utilized to solve contemporary problems, and the algorithms can be applied to big data applications as well.

- Curve fitting

The curve fitting toolbox in MATLAB is useful in analyzing the pattern of data. Once a trend is obtained, which may be a curve or a surface, future trends can be predicted. Additionally, plotting, calculating integrals and derivatives, interpolation, and other related tasks can be performed.

- Control systems

The MATLAB Control Systems toolbox can help to determine the nature of a system. It can provide information on factors such as closed-loop, open-loop, controllability, observability, Bode plot, Nyquist plot, and more. Different controlling techniques like PD, PI, and PID can be easily visualized. Analysis of the system can be performed in either the time or frequency domain.

- Signal Processing

MATLAB offers the advantage of visualizing Signals and Systems and Digital Signal Processing, which are taught in various engineering disciplines. It allows for performing various transforms, such as Laplace and Z, on any given signal and validating theorems. Analysis can be conducted in either the time domain or the frequency domain, and multiple built-in functions are available for this purpose.

- Mapping

Mapping has numerous practical applications across various domains. For instance, in the field of big data, the MapReduce tool is widely used and has numerous real-world applications. Data mapping can be used for a variety of purposes, including theft analysis, financial fraud detection, regression modeling, contingency analysis, prediction techniques in social media, data monitoring, and so on.

- Deep learning

This is a type of machine learning that focuses on specific applications like recognizing speech, detecting financial fraud, and analyzing medical images. To achieve this, tools such as time series, artificial neural networks (ANN), fuzzy logic, or a combination of these can be utilized. MATLAB offers various tools for financial analysis, including identifying factors like profitability, solvency, liquidity, and stability. It also provides tools for evaluating business valuation, capital budgeting, cost of capital, and other related aspects, which can help entrepreneurs make informed decisions.

- Image processing

Algorithms for barcode scanners, facial recognition in selfies (such as beautification or background blurring), and image enhancement are readily available in digital image processing. In addition, it also plays a significant role in the transmission and decoding of data from distant satellites. All necessary algorithms to support these applications are readily available.

- Electric vehicle designing

MATLAB is utilized to model electric vehicles and analyze how they perform when system inputs are altered. It can be used for activities like comparing speed and torque, designing and simulating vehicles, and more.

- Aerospace

This toolbox in MATLAB is used for analyzing the navigation and visualizing flight simulator.

- Computer Vision Toolbox

MATLAB offers various algorithms, functions, and apps that help in designing and evaluating computer vision, video processing, and 3D vision systems. This allows performing tasks such as object detection, feature extraction, tracking, and matching. MATLAB also provides automated calibration workflows for fisheye, single, and stereo cameras [12].

In this project, MATLAB is utilized to create a Libya license plate system that functions automatically. The image processing toolbox enhances and processes license plate images, while the computer vision toolbox labels the dataset and trains the license plate detector and number classifier.

2.4 ModelSim-Altera

"The ModelSim-Altera software is Altera specific and supports behavioral and gatelevel timing simulations and either VHDL or Verilog HDL simulations and test benches for Altera PLDs. ModelSim-Altera was used to design and simulate the traffic system" [14]

2.5 Quartus II

"The Quartus II development software provides a complete design environment for system-on-a-programmable-chip (SOPC) design. Regardless of whether you use a personal computer or a Linux workstation, the Quartus II software ensures easy design entry, fast processing, and straightforward device programming. Quartus II was used to program the FPGA board" [15].

Chapter 3

Related Work

A real-time density-based College of Engineering/University of Tripoli car parking system was proposed and designed using Verilog HDL and implemented on Cyclone IV GX field-programmable gate arrays (FPGA). To enable automatic license plate recognition, machine learning (ML) and image processing algorithms were employed, which were programmed and tested using MATLAB software.

Smart car parking using Arduino Microcontroller project aims at providing a confusion free and easy parking. This project helps the drivers of the cars to park their vehicles with minimum wastage of time with accurate information of the availability of the space to park over Android app. The operator also can collect parking fees efficiently and the drivers can book and pay for their parking space over Android app. It includes an Arduino Uno and Arduino Mega as the microcontroller unit to which the servo motors, LCD, object counter using IC 555 and IC 4026, ultrasonic sensors (HC-05) and IR sensors SR21-IC are interfaced. The LCD displays the availability of the space, the counter keeps the check of the number of cars entering and exiting the parking space, the servo motor helps as gate for the entry and exit of the cars. The ultrasonic sensors detect the availability of the parking space [16].

The Libyan license plates recognition project utilizes a support vector machine, and it consists of three primary stages. The first stage is plate detection, which is accomplished using vertical and horizontal histograms. The second stage is character segmentation, which employs a connected-component labeling algorithm. Finally, optical character recognition (OCR) is performed using support vector machines (SVMs) [17].

Chapter 4

Parking System

4.1 System overview

The "College of Engineering/University of Tripoli Smart Car Parking System Based on Plate Recognition" project has 142 slots focuses on the hardware implementation using Verilog and FPGA. The system utilizes an array of IR sensors to divide the parking area into four section slots, enhancing organization. By incorporating plate recognition technology, the system can accurately identify vehicles; the system can calculate parking duration for billing purposes. Thorough testing has been conducted to ensure the system's reliability and effectiveness. The project aims to provide users at the College of Engineering/University of Tripoli with an automated and well-organized parking experience.

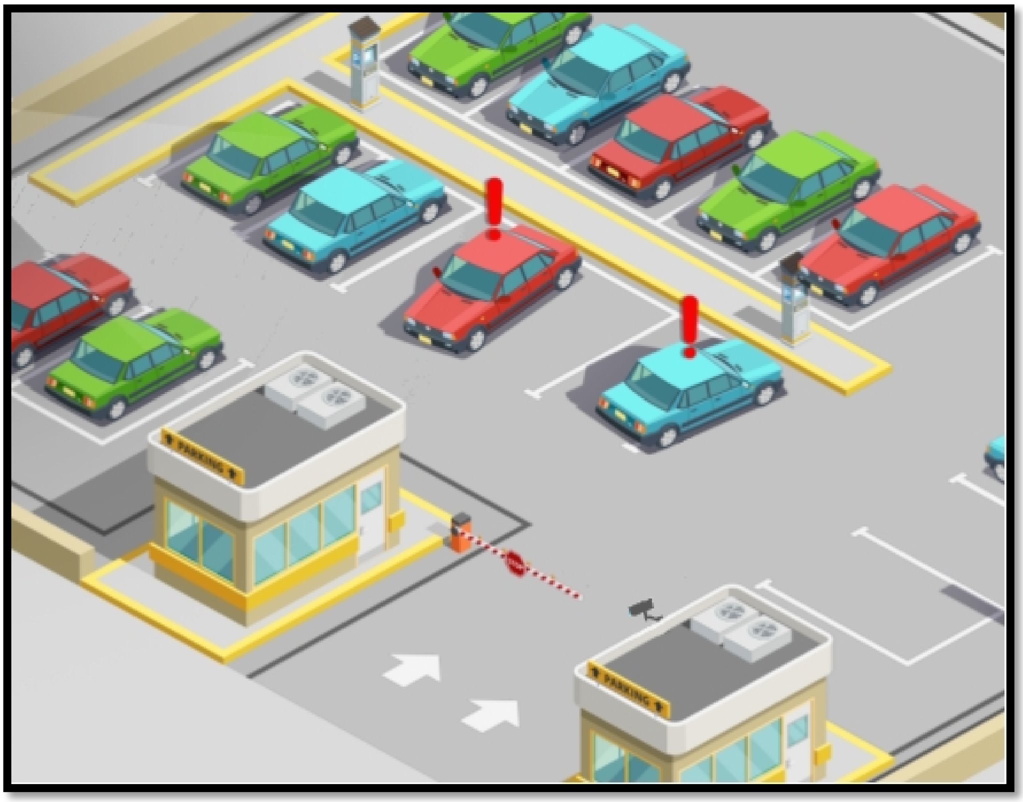

Figure 1. System structure overview

4.2 System Algorithm

The implemented car parking system operates as follows: When a vehicle enters the parking lot, the system checks for the availability of parking spaces. If a space is found to be vacant, the entrance gate is activated, allowing the vehicle to access the parking area. Facilitating the transmission and reception of slot availability information, the RF module plays a crucial role in this process. It enables the system to receive real-time updates regarding the availability of parking slots and triggers the illumination of LEDs to provide visible indications to the users.

This car parking system is designed to accommodate various categories of individuals, including staff, faculty members, students, and visitors. Each category is assigned a specific code and unique identification (ID) for accessing the parking areas. The system encompasses four distinct areas within the parking facility that are accessible to all authorized individuals falling under these categories.

To visually represent the operational flow of the car parking system, a comprehensive flowchart has been developed. This flowchart employs oval-shaped boxes to symbolize the system's outputs, which are influenced by both previous states and current inputs. The accompanying figure visually illustrates this flowchart, providing a clear depiction of how the proposed model operates.

The system flowchart commences with an initial verification process at the entrance gate. During this stage, the system verifies the availability of parking spaces by examining the ID password provided by the user. This verification process is tailored to different categories, namely staff, faculty members, students, and visitors, each possessing a designated code and ID. To determine the availability of spaces in each parking area, the system cross-references the entered ID password or code with the corresponding category assigned to the specific parking lot. In instances where no available spaces are found, the entrance gate remains closed, preventing access to the parking facility.

Furthermore, the car parking system incorporates an additional validation step to ensure the accuracy and validity of the entered password. This validation process occurs at the entrance gate and is triggered upon the submission of an ID password. If the system identifies an invalid password, it generates an error signal, prompting the gate to remain closed until a valid password is provided by the user.

In situations where staff members input their assigned ID password, the space availability check extends to the student parking lots area. This unique feature prioritizes staff members by granting them access to parking spaces allocated for students in cases where no other spaces are available. This ensures that staff members have alternative parking options within the facility, enhancing convenience and accessibility.

Within the car parking system, specific provisions are in place to accommodate faculty members and students. When an individual from either category enters their respective password, the system operates to allocate a designated lot area with available parking spaces. By analyzing the entered password, the system intelligently determines which lot area can provide a free parking spot for the faculty member or student. This streamlined process ensures efficient utilization of parking resources and allows individuals to swiftly proceed to the assigned lot area.

Furthermore, when it comes to exiting the parking facility, a specific code needs to be entered to initiate the departure process. This code serves as a validation mechanism to ensure that the individual leaving the parking area is authorized to do so. By requiring the code for departure, the system enhances security measures and prevents unauthorized access to the parking facility.

It is important to note that these measures are implemented for faculty members and students, emphasizing the system's tailored approach to meet the parking needs of these two categories. By providing the free number lot information based on the entered password and requiring a departure code, the system effectively manages parking allocation and ensures a secure and controlled exit process for faculty members and students.

It is important to note that the suggested car parking system implements a payment policy exclusively for visitors. Staff, faculty members, and students are exempted from any vehicle hosting fees. However, visitors are required to pay for each hour of reservation in the parking lot. To facilitate this payment process, a 7-segment display integrated into the system showcases the amount due after the visitor's vehicle is detected at the exit point. Subsequently, visitors have the option to make their payment at the gate. Upon completing the payment, the gate promptly opens, enabling them to exit the parking facility without delay.

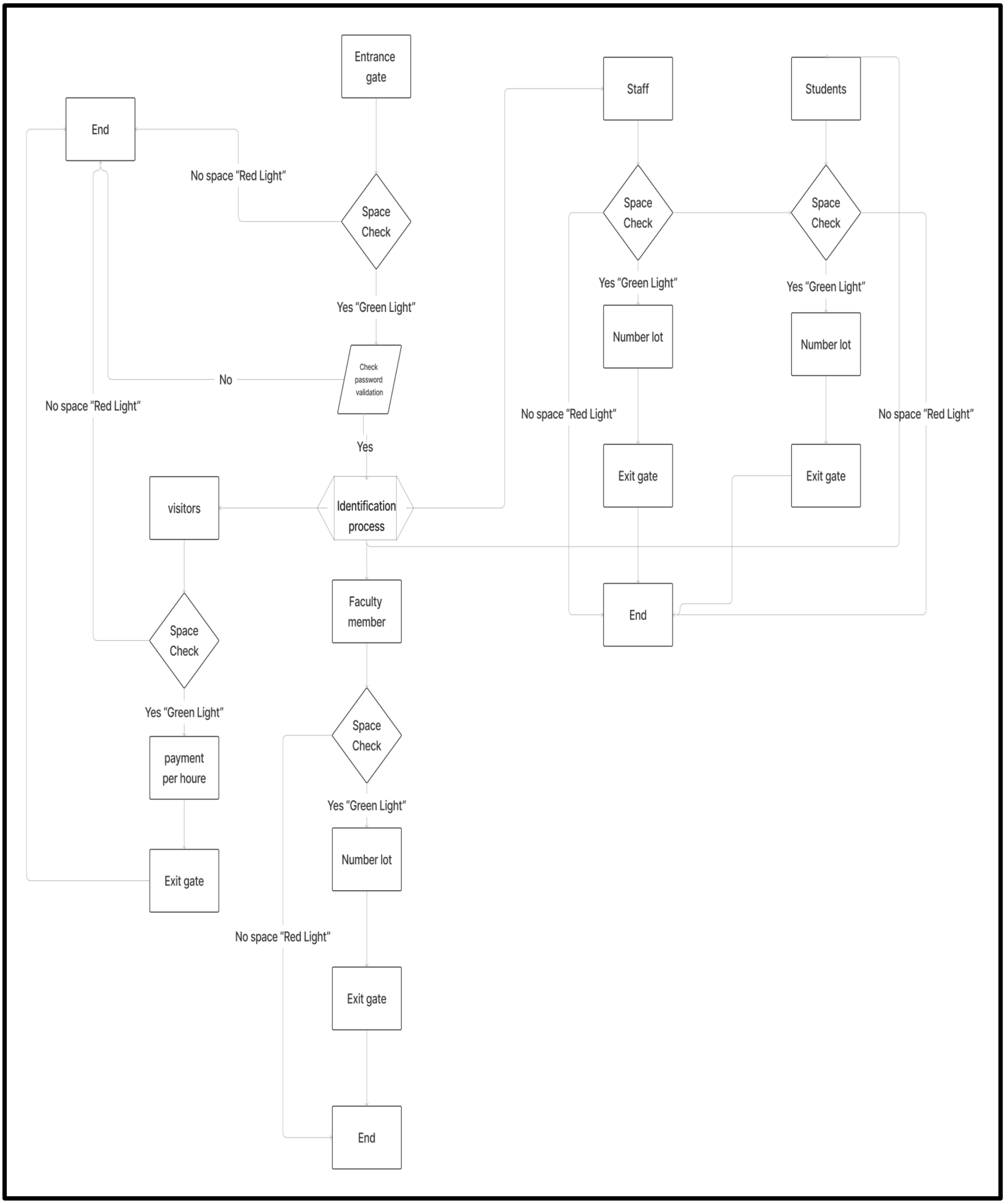

Figure 2.System structure flowchart

4.3 Features offered by the system

1. Efficient Use of Time and Systematic Parking: The system aims to optimize time utilization by curbing wastage in unproductive areas like vehicle parking. It ensures systematic parking by indicating available vacancies, considering adjacent spaces, and calculating distances to obstacles.
2. Enhanced Security and Safe Parking: The system prioritizes security by incorporating password authentication and providing a safe parking environment, reducing risks associated with parking.
3. Integration with Public Services: The parking system seamlessly integrates with other public utilities, allowing effective operation and coordination within a parking network.
4. FPGA Implementation and Flexibility: The system utilizes a reconfigurable FPGA architecture, providing flexibility, programmability, and ease of modification for efficient parking management.
5. Real-time Density-based Traffic Signal Integration: The parking system integrates with a real-time density-based system for traffic signals. It dynamically adjusts the green signal entrance time based on the density of vehicles, ensuring optimal utilization of available parking spaces.
6. The parking system implemented at Tripoli University College of Engineering gives priority to staff vehicles by allowing them to check into student slots if the staff parking area is full or unavailable.
7. Utilization of Parking System as a Source of Income for Tripoli University College of Engineering Faculty: The parking system implemented at Tripoli University College of Engineering offers the additional benefit of generating income for the faculty. By leveraging the parking facilities, the system contributes to the financial well-being of the institution. The generated income can be utilized for the betterment and improvement of the college, aligning with the same approach as the previously mentioned features.

4.4 Results and discussions

To verify the working of the system, we have used ModelSim-Intel-ALTER to obtain the timing diagram. In the timing diagram sensor_lot1 is for staff members, sensor_lot2 is for faculty members, sensor_lot3 for students, and sensor_lot4 is for visitors. Each are connected with array of IR sensors for each slot also sensor_ entrance is for in gate and sensor_exit signal is for the out gate. Each connected with green_LED when there is available lot and red if there is no free slot to check in.

The parking system utilizes a password-based access control mechanism, where each category (staff, faculty members, students, and visitors) is assigned a unique code. The codes assigned to each category are as follows: staff (1019, lots 1-45), faculty members (750, lots 46-91), students (317, lots 92-135), and visitors (lots 136-142). To gain access to the parking area, the entered code must match the assigned code for the respective category. In case of an incorrect code, the system activates an Error signal (Error = 1) until a valid code is provided. Moreover, if there are no available parking spaces, the entrance gate remains closed (gate_entrance = 0). On the other hand, for exiting the parking area, a specific code (code_exit) must be entered. Only when the correct code_exit is provided, the exit gate (gate_exit) opens (gate_exit = 1), allowing vehicles to leave.

Staff members have the privilege to access and utilize student slots temporarily if there are no available staff parking spaces. They can occupy student slots for a specific duration. However, once a staff parking space becomes available, an alert signal is triggered (alert = 1), indicating that staff members should check in to the designated staff parking lot. If there are no available parking slots, the gate will remain closed, and other categories of users will not have access to any parking slots.

The default number of hours assigned to each visitor's parking lot is set to 1 and 5LYD for each hour host. Visitors have the option to check in by presenting a provided ID. However, the exit gate will only open for them to leave when the payment check (check_pay) is set to 1.

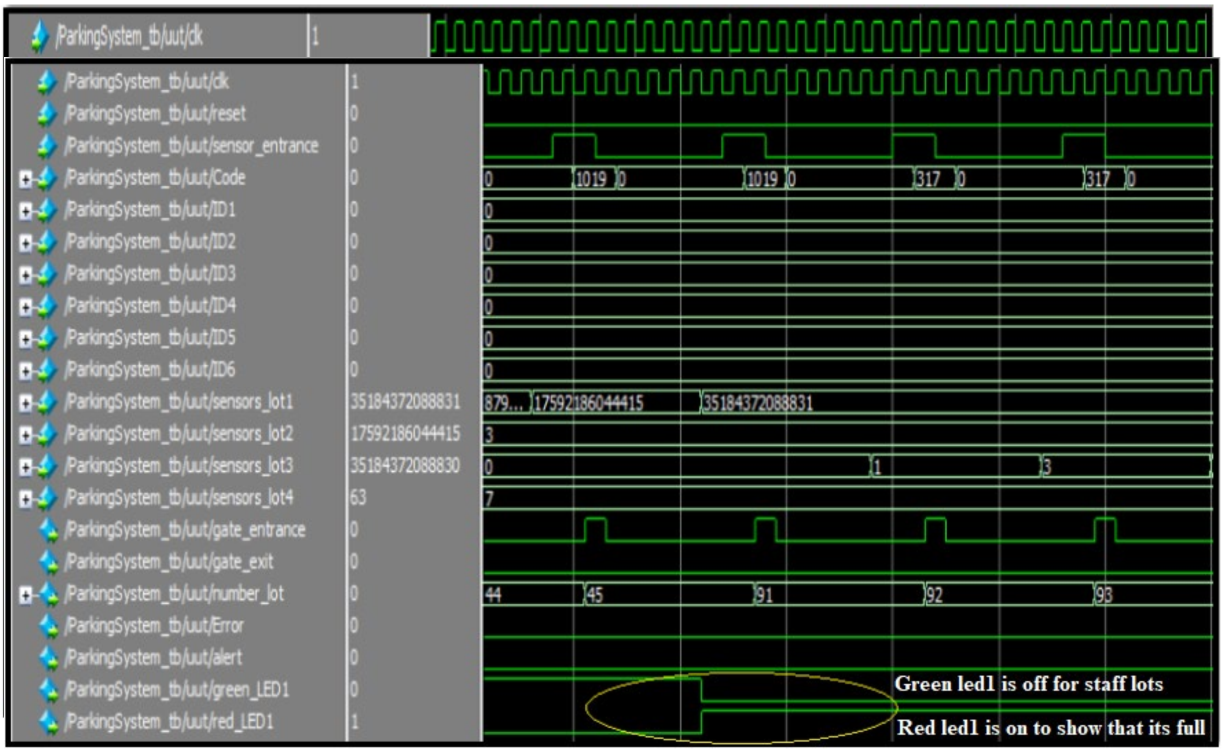

Figure 3 .The timing diagram for the real-time density-based parking system with gate_entrance signal.

Figure 4. The timing diagram for the real-time density-based parking system with red_LED1 signal.

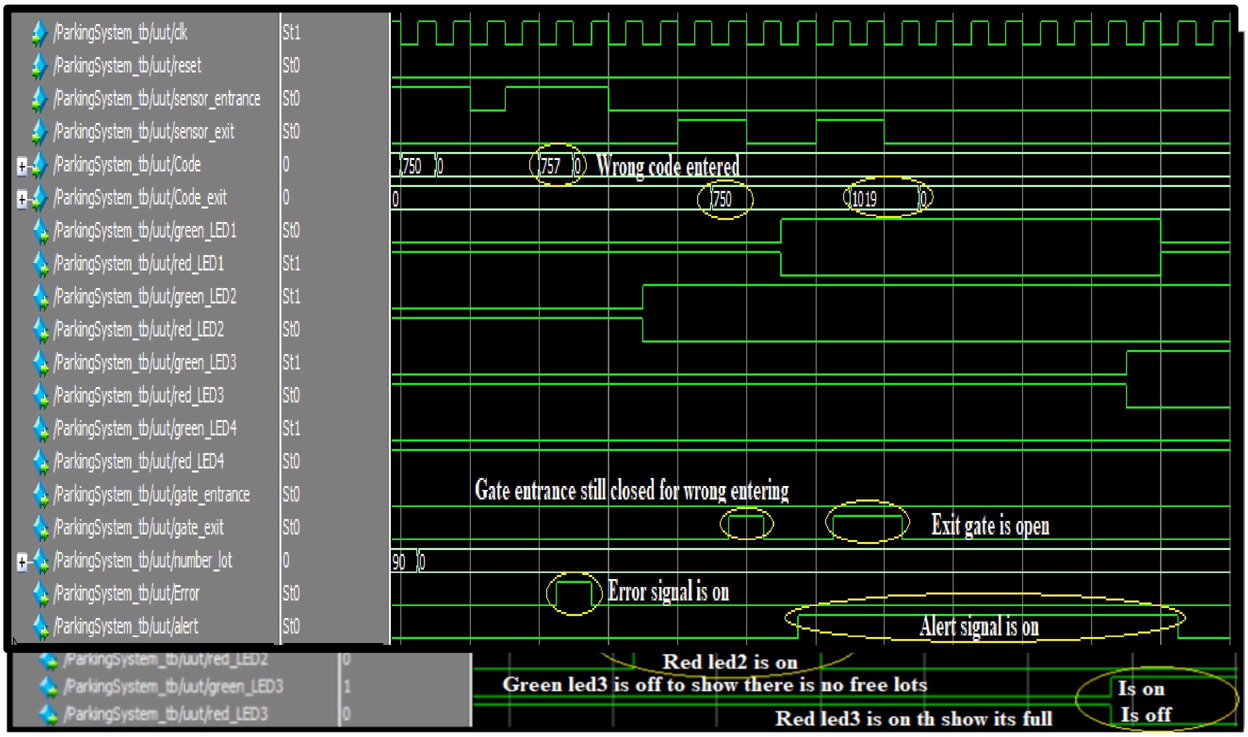

Figure 6. The timing diagram for the real-time density-based parking system with red_LED2 and red_LED3 signals.

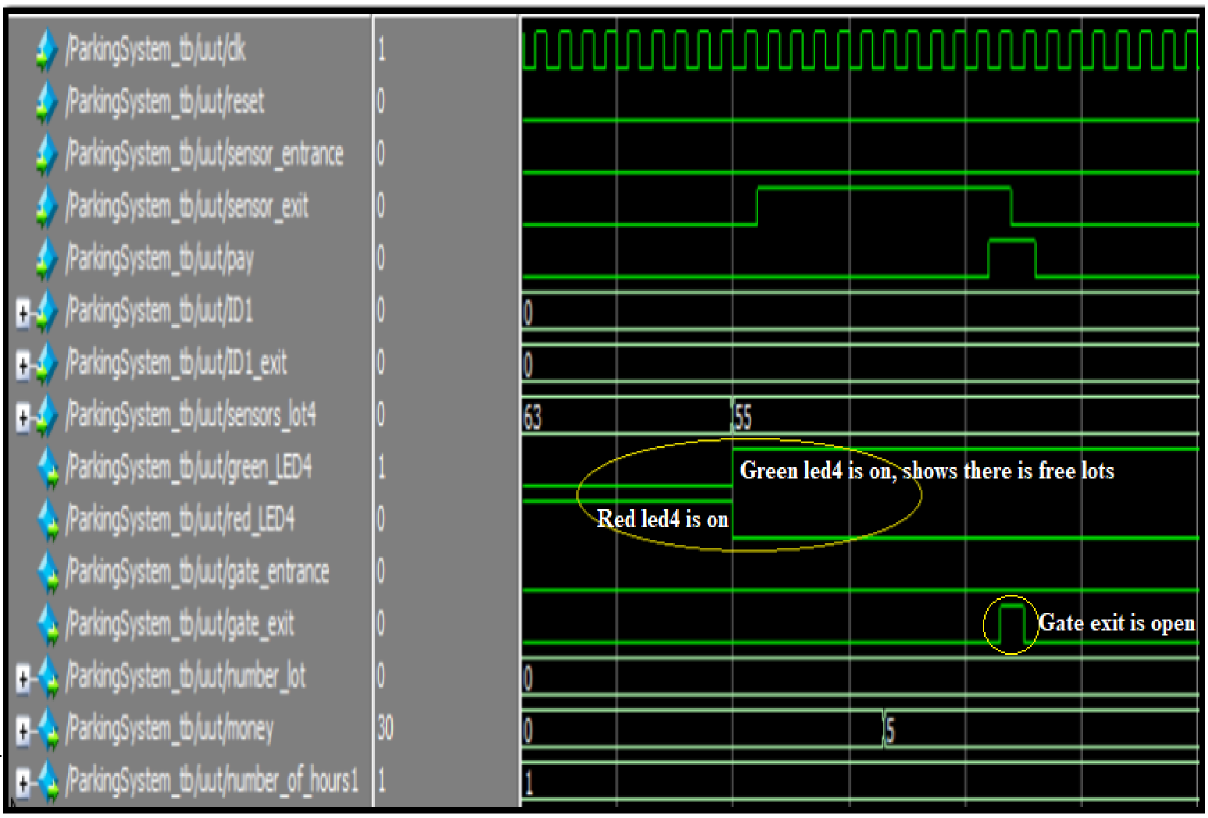

Figure 7. The timing diagram for the real-time density-based parking system with red_LED4 signal.

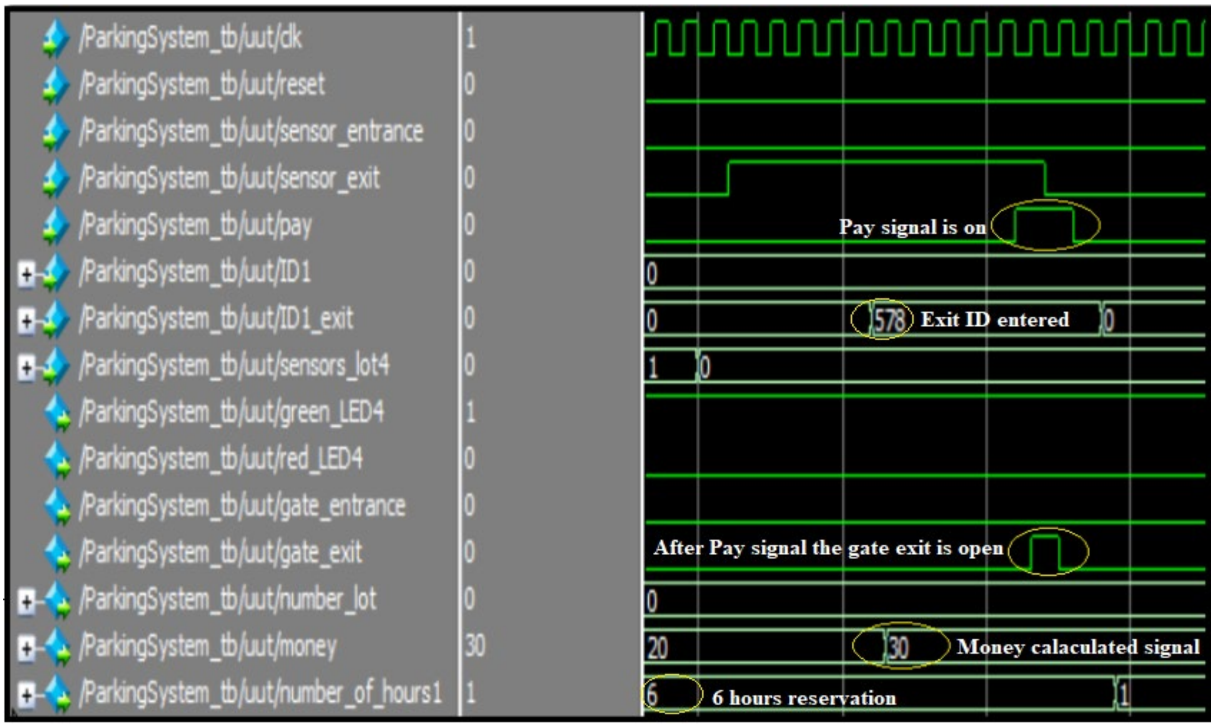

Figure 8. The timing diagram for the real-time density-based parking system with money,pay and gate_exit signals for the first visitor leave.

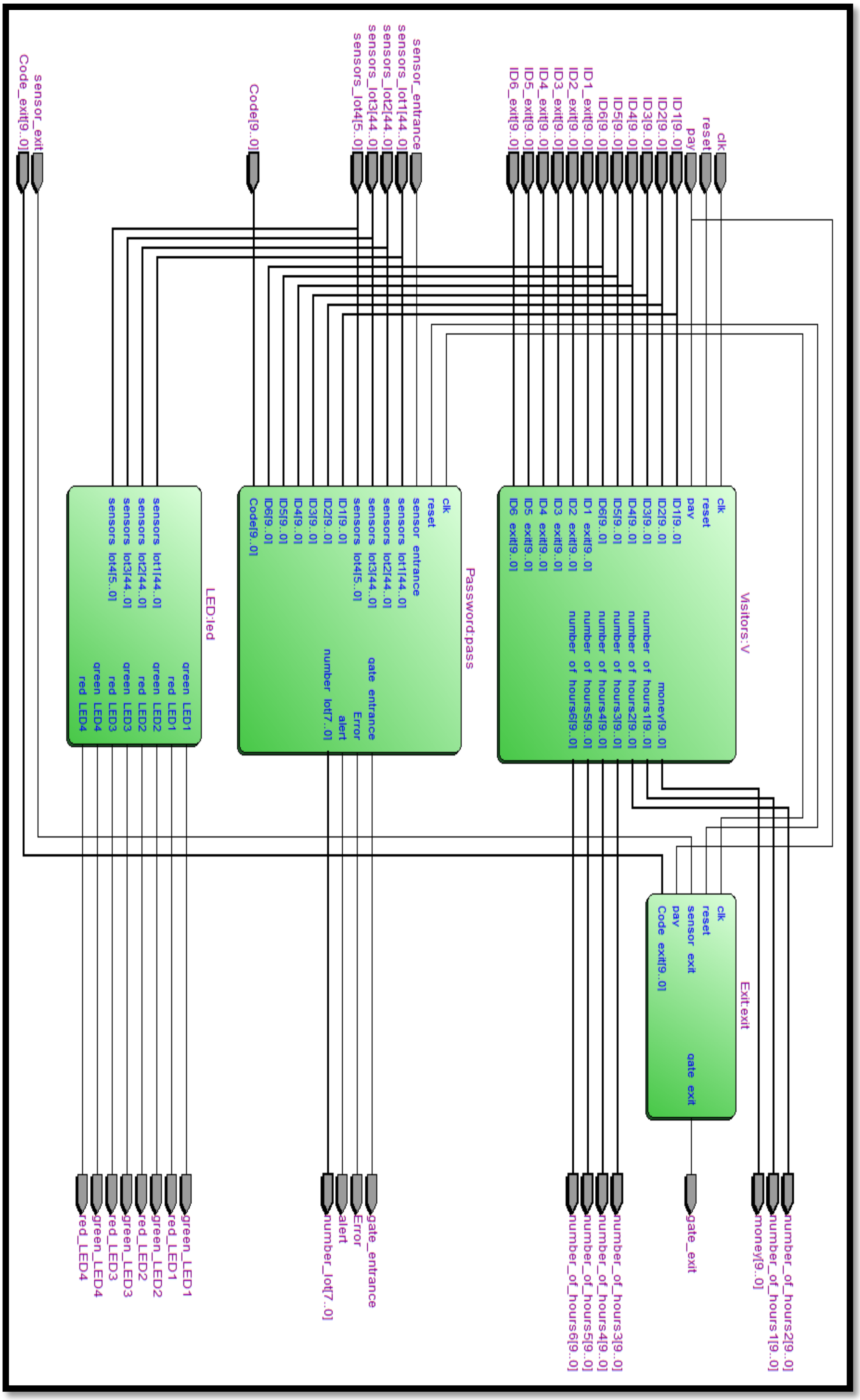

Figure 9. Register Transfer Level RTL viewer

Chapter 5

License Plate Recognition

The automatic license plate system that is suggested is comprised of four components:

1. Detection and image acquisition.
2. License plate localization.
3. Image pre-processing.
4. Extract the license number.
5. License plate number recognition.

5.1 Detection and image acquisition.

The camera captures the vehicle image then the image is sent to the license plate recognition.

5.2 License plate localization.

The algorithm uses machine learning to locate license plate regions in images using the ACF method. The detector is trained on a dataset of Libyan license plates, allowing it to detect plates from varying angles and distances, including skewed plates. The detector is trained on 2899 images and tested on 742 images, achieving an average precision of 0.9.

Figure 10. Detected license plate.

5.3 Image pre-processing

Once the license plate region is identified, it is cropped and undergoes various image processing techniques to make it ready for digit recognition and extraction.

1. Extract the license plate region.

During this stage, the region of interest in the image is extracted by utilizing the "imcrop" function in MATLAB, which crops the “BoundingBox” produced by the license plate detector.

Figure 11. Cropped license plate.

2. Convert the license plate to gray level.

Figure 12. Gray level of license plate.

3. Convert the license plate to binary level.

Figure 13. Binary image of license plate.

4. Remove the black border.

To prevent errors in extracting the license number, the black border is removed using the following method:

a) Perform two-dimensional median filtering on plate images.

b) Perform hole Filling.

c) NOT (XOR (result of median filtering, result of hole filling).

Figure 14. License image after removing black border.

5. Remove small objects.

The license plate is adjusted to a uniform size of [145 × 520] in this process. Subsequently, objects that are smaller than 390 pixels in size are eliminated, since they cannot be Libya words or digits.

Figure 15. License plate image after removing small objects.

5.4 Extraction of the license numbers

The license plate number is extracted via connected-component. A red rectangle is drawn around the identified bounding box on the output graph. From top to bottom and left to right, the connected-component labeling begins extracting objects from an image. The connected-component extracting method is appropriate given that Libyan license plates are rectangular in shape and include all of their numbers in a single line.

Figure 16. Extracted license number.

5.5 License plate recognition

The image properties are extracted using the "regionprops" function, which returns various properties of the connected components in the input image. Specifically, the "BoundingBox" and 'Image' properties are extracted and stored in "imgProp". Next, a series of 10 grayscale digit images (0-9) are loaded into the script. The variable "cnt" is initialized to 0 and an empty string "out" is created. The "for" loop then iterates through the components in "imgProp". For each component, its bounding box dimensions are checked to ensure they fall within a certain range of values. If the bounding box dimensions are within the specified range, the corresponding image is isolated and resized to 40x20 pixels. This resized image is then saved as "temp.bmp" and loaded back in as "imgTest".

The script then compares the "imgTest" image against each of the grayscale digit images using "imabsdiff" function. The digit image that has the smallest difference (sum of absolute differences) with "imgTest" is selected as the correct digit and added to 'out'. The loop then moves on to the next component in "imgProp" and repeats the process until all components have been processed.

Figure 17. Result of license plate recognition.

5.6 Saving license plate number

The system automatically saves the recognized license plate number in an Excel sheet for security purposes.

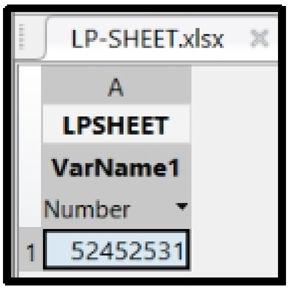

Figure 18. Saved result of license plate number.

5.7 Results and discussions

The MATLAB-based Libyan license plate recognition system employs an ACF license plate detector for the license plate localization step, which was trained on 2899 images and tested on 742 images. The license plate detector was evaluated on the test data, and an average precision of 0.9 was achieved, as illustrated in Figure .number. The pre-processing step significantly improved the accuracy of the license plate number extraction step on the tested images.

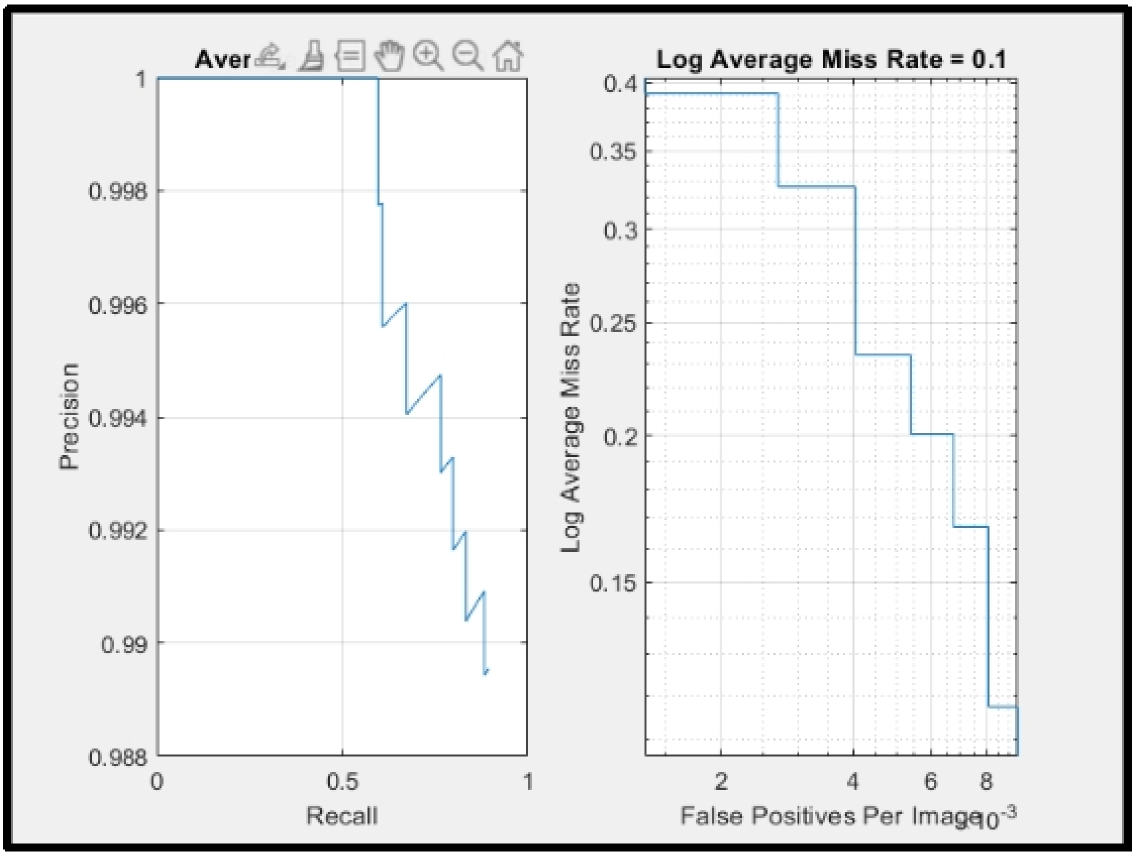

Figure 19. Evaluation of license plate detector.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The objective of this project is to create, test, and implement a real-time parking lot system for the College of Engineering/University of Tripoli. The system includes a license plate recognition feature to improve security and reduce traffic congestion on roads. This project's significant contribution is the integration of a Libyan license plate recognition system into the parking system. The main advantages of the system are its capability to generate revenue, organizing the parking slots into three areas one for the students and two for staff and visitors with prioritizing the staff and students of the College of Engineering/University of Tripoli. The parking system module was tested, and the design was verified using software tools from Intel-Altera. Afterward, the system was implemented and tested on the Cyclone IV GX: EP4CGX150DF31C8 FPGA evaluation platform to confirm its correct functioning in hardware. The purpose of the license plate recognition system in Libya is to enhance security through the application of machine learning and image processing techniques. The software for recognizing license plates has been developed and evaluated using the MATLAB software tool.

6.2 Future Work

the proposed system can be improved by connect it with all Tripoli University parking areas and linked them with a mobile application benefited for the general category of people so that they can choose the required park lot and learn about its fees and available places.

DE2i-150 FPGA board

In this project, the Terasic DE2i-150 board from Intel Altera, which is a type of integrated circuit composed of semiconductor material and can be reprogrammed or configured by the user after purchase, was utilized. This is in contrast to relying solely on the original equipment manufacturer (OEM) [17].

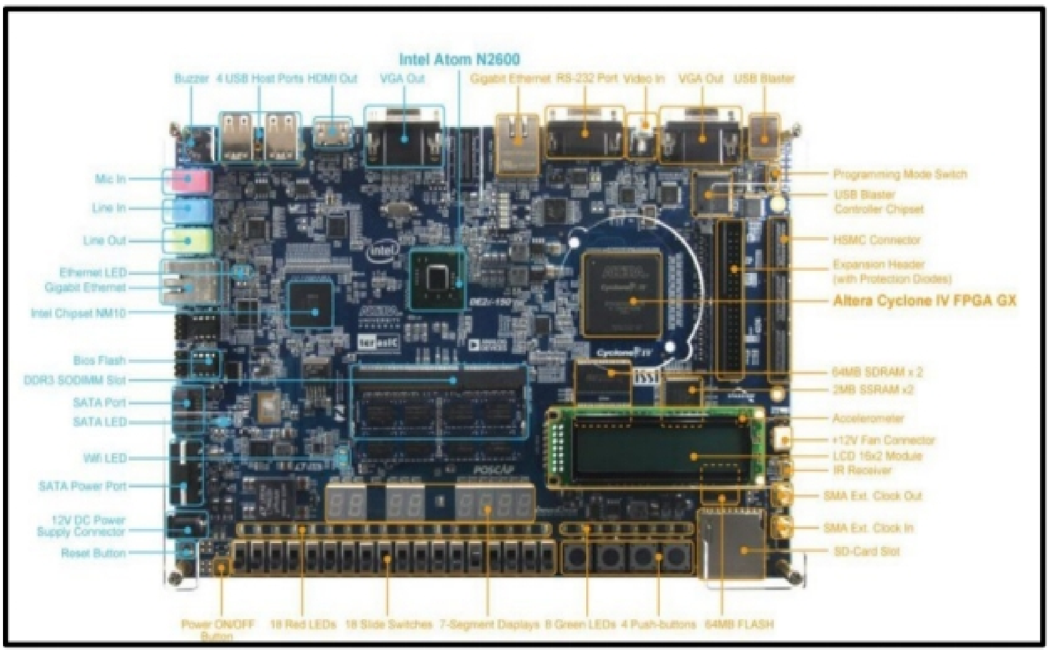

Figure 20.The DE2i-150 board top view.

"The DE2i-150 board has many features that allow users to implement a wide range of designed circuits, from simple circuits to various multimedia projects.

The following hardware (FPGA System) is provided on the DE2i-150 board:

- Altera Cyclone® IV 4CX150 FPGA device
- Altera Serial Configuration device – EPCS64
- USB Blaster (on board) for programming; both JTAG and Active Serial (AS) programming
- Modes are supported
- Two 2MB SSRAM
- Two 64MB SDRAM
- 64MB Flash memory

- SD Card socket
- 4 Push-buttons
- 18 Slide switches
- 18 Red user LEDs
- 9 Green user LEDs
- 50MHz oscillator for clock sources
- VGA DAC (8-bit high-speed triple DACs) with VGA-out connector
- TV Decoder (NTSC/PAL/SECAM) and TV-in connector
- Gigabit Ethernet PHY with RJ45 connectors
- RS-232 transceiver and 9-pin connector
- IR Receiver
- 2 SMA connectors for external clock input/output
- One 40-pin Expansion Header with diode protection
- One High-Speed Mezzanine Card (HSMC) connector”
- 16x2 LCD module [18].

Code[2]	Input	PIN_P30	6	B6_N2	2.5 V (default)		16mA (default)		
Code[1]	Input	PIN_R30	6	B6_N2	2.5 V (default)		16mA (default)		
Code[0]	Input	PIN_T28	6	B6_N2	2.5 V (default)		16mA (default)		
Code_exit[2]	Input	PIN_N26	6	B6_N2	2.5 V (default)		16mA (default)		
Code_exit[1]	Input	PIN_R26	6	B6_N2	2.5 V (default)		16mA (default)		
Code_exit[0]	Input	PIN_R29	6	B6_N2	2.5 V (default)		16mA (default)		
Error	Output	PIN_M22	6	B6_N1	2.5 V (default)		16mA (default)	2 (default)	
ID1	Input	PIN_M26	6	B6_N2	2.5 V (default)		16mA (default)		
ID1_exit	Input	PIN_N25	6	B6_N2	2.5 V (default)		16mA (default)		
alert	Output	PIN_M21	6	B6_N1	2.5 V (default)		16mA (default)	2 (default)	
clk	Input	PIN_AJ16	4	B4_N2	2.5 V (default)		16mA (default)		
gate_entrance	Output	PIN_AA25	5	B5_N2	2.5 V (default)		16mA (default)	2 (default)	
gate_exit	Output	PIN_AB25	5	B5_N2	2.5 V (default)		16mA (default)	2 (default)	
green_LED1	Output	PIN_F27	6	B6_N0	2.5 V (default)		16mA (default)	2 (default)	
green_LED2	Output	PIN_F26	6	B6_N0	2.5 V (default)		16mA (default)	2 (default)	
green_LED3	Output	PIN_W26	5	B5_N0	2.5 V (default)		16mA (default)	2 (default)	
green_LED4	Output	PIN_Y22	5	B5_N1	2.5 V (default)		16mA (default)	2 (default)	
number_lot[2]	Output	PIN_J25	6	B6_N0	PIN_J25	2.5 V (default)		16mA (default)	2 (default)
number_lot[1]	Output	PIN_AA22	5	B5_N1	PIN_AA22	2.5 V (default)		16mA (default)	2 (default)
number_lot[0]	Output	PIN_Y25	5	B5_N2	PIN_Y25	2.5 V (default)		16mA (default)	2 (default)
number_of_hours1[5]	Output	PIN_L25	6	B6_N1	PIN_L25	2.5 V (default)		16mA (default)	2 (default)
number_of_hours1[4]	Output	PIN_K24	6	B6_N0	PIN_K24	2.5 V (default)		16mA (default)	2 (default)
number_of_hours1[3]	Output	PIN_M25	6	B6_N2	PIN_M25	2.5 V (default)		16mA (default)	2 (default)
number_of_hours1[2]	Output	PIN_N21	6	B6_N1	PIN_N21	2.5 V (default)		16mA (default)	2 (default)
number_of_hours1[1]	Output	PIN_N24	6	B6_N2	PIN_N24	2.5 V (default)		16mA (default)	2 (default)
number_of_hours1[0]	Output	PIN_P21	6	B6_N1	PIN_P21	2.5 V (default)		16mA (default)	2 (default)
pay	Input	PIN_J26	6	B6_N1	PIN_J26	2.5 V (default)		16mA (default)	
red_LED1	Output	PIN_T23	5	B5_N0	PIN_T23	2.5 V (default)		16mA (default)	2 (default)
red_LED2	Output	PIN_T24	5	B5_N0	PIN_T24	2.5 V (default)		16mA (default)	2 (default)
red_LED3	Output	PIN_V27	5	B5_N0	PIN_V27	2.5 V (default)		16mA (default)	2 (default)
red_LED4	Output	PIN_W25	5	B5_N0	PIN_W25	2.5 V (default)		16mA (default)	2 (default)
reset	Input	PIN_AA26	5	B5_N2	PIN_AA26	2.5 V (default)		16mA (default)	

Figure 21. Pins assignment.

seg1[6]	Output	PIN_F14	8	B8_N0	PIN_F14	2.5 V (default)		16mA (default)	2 (default)
seg1[5]	Output	PIN_D16	8	B8_N0	PIN_D16	2.5 V (default)		16mA (default)	2 (default)
seg1[4]	Output	PIN_F16	8	B8_N0	PIN_F16	2.5 V (default)		16mA (default)	2 (default)
seg1[3]	Output	PIN_F11	8	B8_N1	PIN_F11	2.5 V (default)		16mA (default)	2 (default)
seg1[2]	Output	PIN_G11	8	B8_N1	PIN_G11	2.5 V (default)		16mA (default)	2 (default)
seg1[1]	Output	PIN_E12	8	B8_N1	PIN_E12	2.5 V (default)		16mA (default)	2 (default)
seg1[0]	Output	PIN_E15	8	B8_N0	PIN_E15	2.5 V (default)		16mA (default)	2 (default)
seg2[6]	Output	PIN_G10	8	B8_N2	PIN_G10	2.5 V (default)		16mA (default)	2 (default)
seg2[5]	Output	PIN_J9	8	B8_N2	PIN_J9	2.5 V (default)		16mA (default)	2 (default)
seg2[4]	Output	PIN_G12	8	B8_N1	PIN_G12	2.5 V (default)		16mA (default)	2 (default)
seg2[3]	Output	PIN_F12	8	B8_N1	PIN_F12	2.5 V (default)		16mA (default)	2 (default)
seg2[2]	Output	PIN_G13	8	B8_N0	PIN_G13	2.5 V (default)		16mA (default)	2 (default)
seg2[1]	Output	PIN_B13	8	B8_N0	PIN_B13	2.5 V (default)		16mA (default)	2 (default)
seg2[0]	Output	PIN_G14	8	B8_N0	PIN_G14	2.5 V (default)		16mA (default)	2 (default)
seg3[6]	Output	PIN_F10	8	B8_N2	PIN_F10	2.5 V (default)		16mA (default)	2 (default)
seg3[5]	Output	PIN_F4	8	B8_N2	PIN_F4	2.5 V (default)		16mA (default)	2 (default)
seg3[4]	Output	PIN_F6	8	B8_N2	PIN_F6	2.5 V (default)		16mA (default)	2 (default)
seg3[3]	Output	PIN_AG30	5	B5_N2	PIN_AG30	2.5 V (default)		16mA (default)	2 (default)
seg3[2]	Output	PIN_F7	8	B8_N2	PIN_F7	2.5 V (default)		16mA (default)	2 (default)
seg3[1]	Output	PIN_G7	8	B8_N2	PIN_G7	2.5 V (default)		16mA (default)	2 (default)
seg3[0]	Output	PIN_G8	8	B8_N2	PIN_G8	2.5 V (default)		16mA (default)	2 (default)
sensor_entrance	Input	PIN_V28	5	B5_N0	PIN_V28	2.5 V (default)		16mA (default)	
sensor_exit	Input	PIN_U30	5	B5_N0	PIN_U30	2.5 V (default)		16mA (default)	
sensors_lot1	Input	PIN_V21	5	B5_N1	PIN_V21	2.5 V (default)		16mA (default)	
sensors_lot2	Input	PIN_C2	8	B8_N2	PIN_C2	2.5 V (default)		16mA (default)	
sensors_lot3	Input	PIN_AB30	5	B5_N1	PIN_AB30	2.5 V (default)		16mA (default)	
sensors_lot4	Input	PIN_U21	5	B5_N1	PIN_U21	2.5 V (default)		16mA (default)	
data_bus[7]	Unknown	PIN_AF4	3	B3_N2		2.5 V (default)		16mA (default)	
data_bus[6]	Unknown	PIN_AH4	3	B3_N2		2.5 V (default)		16mA (default)	
data_bus[5]	Unknown	PIN_AE3	3	B3_N2		2.5 V (default)		16mA (default)	
data_bus[4]	Unknown	PIN_AH2	3	B3_N2		2.5 V (default)		16mA (default)	
data_bus[3]	Unknown	PIN_AE5	3	B3_N2		2.5 V (default)		16mA (default)	
data_bus[2]	Unknown	PIN_AH3	3	B3_N2		2.5 V (default)		16mA (default)	
data_bus[1]	Unknown	PIN_AF3	3	B3_N2		2.5 V (default)		16mA (default)	
data_bus[0]	Unknown	PIN_AG4	3	B3_N2		2.5 V (default)		16mA (default)	
en	Unknown	PIN_AF4	3	B3_N2		2.5 V (default)		16mA (default)	
rw	Unknown	PIN_AJ3	3	B3_N2		2.5 V (default)		16mA (default)	
rs	Unknown	PIN_AG3	3	B3_N2		2.5 V (default)		16mA (default)	
<<new node>>									

Figure 22. Pins assignment.

Reference

[1] Gao, Z., Tan, Y., Zheng, N., Zhao, C., & Wang, H. (2019). A deep learning framework for license plate detection and recognition in complex conditions. IEEE Transactions on Intelligent Transportation Systems, 21(2), 623-634.

[2] Alsaraireh, M., Hawalah, A., & Jararweh, Y. (2021). "A machine learning-based intelligent parking system." SN Computer Science, 2(2), 1-17.

[3] Chao, H., Gao, J., Sun, G., Wu, Y., & Zhang, J. (2020). "Smart parking system using an intelligent FPGA." IEEE Transactions on Industrial Informatics, 16(6), 4200-4212.

[4] Wang, G., Chen, W., Zhang, W., Cao, D., & Ma, W. Y. (2018). Deep learning for smart vehicles and intelligent transportation: A survey. IEEE Transactions on Intelligent Transportation Systems, 20.

[5] Elhamshary, M. R., & Taha, M. (2020). "Automatic license plate recognition system based on combined classifiers and OCR techniques." Pattern Analysis and Applications, 23(1), 281-295.

[6] Khan, M. R., Hoque, M. M., Ahmed, T., Ahmed, S., & Rahman, M. A. (2021). "A smart parking system using computer vision and internet of things." SN Computer Science, 2(6), 1-14.

[7] Shah, A., & Ali, W. (2021). "License plate recognition using YOLOv4 and OpenCV." In Proceedings of the 2021 3rd International Conference on Computer Science and Artificial Intelligence (pp. 64-68). ACM.

[8] J. R. Aworemi, I. A. Abdul-Azeez, A. J. Oyedokun, and J.O.Adewoye, "A study of the causes, effects and ameliorative measures of road traffic congestion in Lagos Metropolis," European Journal of Social Sciences, vol. 11, no. 1, pp. 119-128, 2009.

[9] https://robocraze.com/blogs/post/ir-sensor-working

[10] https://www.youngwonks.com/blog/What-is-an-infrared-sensor

[11] https://media.neliti.com/media/pubications/263281-smart-car-parking-using-arduino-microcon-d1039137.pdf

[12] https://en.wikipedia.org/wiki/MATLAB.

[13] https://www.geeksforgeeks.org/applications-of-matlab

[14] https://home.engineering.iastate.edu/~zzhang/courses/cpre581-f05/resources/modelsim.pdf

[15]https://www.intel.com/content/www/us/en/programmable/quartushelp/13.0/mergedProjects/quartus/gl_quartus_welcome.htm

[16] A. M. Hassan, S. A.Ghoul, and A. A.Alkabir. "Libyan Vehicle License Plate Recognition with Support Vector Machine," Al-Mukhtar Journal of Sciences, vol. 37, pp. 1--13, 2022.

[17] DE2i-150_FPGA_System_manual

[18] https://www.arrow.com/en/research-and-events/articles/fpga-basics-architecture applications-and-uses