Final Documentation

TEAM F

28.06.2024

1 Team members

Kashif Raza
Ammar Imran Khan
Hammad Karamat

2 Introduction

This paper focuses on implementation of parking place monitoring system on an FPGA, which is capable of easing parking management with better efficiency and accuracy. In this paper, a system for parking place monitoring using ultrasonic and infrared sensors is described, which works on real-time status updating of parking spots. The data from the sensors is processed by the FPGA to identify whether the place is occupied or empty and this information is made available through multiple 7-segment displays the data processed from ultrasonic sensor measures the distance of the car edge to the border of the parking spot to make aware the use, with the sound of buzzer, for a possible collision.

This project makes a strong, growing, and fast answer to today's parking needs. By using FPGA tech, the system gives top speed and easy changes to meet the changing needs of parking control. The use of many sensors and instant data handling boosts the true and steady results of the parking watch system, making it a key aid for managing places.

Use of Field-programmable gate arrays (FPGAs) and VHSIC hardware description language (VHDL) are strong ideas for the development of a parking place monitoring system. FPGAs are very instrumental in this case because they possess high performance levels and parallel computing abilities which are required in real-time systems like this one. They do this by processing many sensor inputs at once, thus ensuring fast and accurate detection of the state of parking spaces. Furthermore, these FPGAs have reprogrammable hardware which makes them easily adaptable to new conditions or requirements that may necessitate changes to be made to the whole hardware design.

The main benefit of having low latency FPGAs is that it allows for deterministic and immediate responses necessary for real-time systems. With real time data transfer capabilities, it allows our project to be evolved for IOT applications for future possibilities. Real time data parsing allows it to beneficial for network related applications. It’s important to note that collisions prevention systems also depend on this feature so as ensure that collision don’t occur with such kind of parking involved. The other advantage with these devices is their scalability since they can handle multiple activities concurrently hence suitable for expanding the system into a more robust parking space monitoring platform. This means that no matter how big the parking lot becomes, it will still be efficient enough because it can expand with its users. Equally important is that these FPGAs are programmable allowing for customization through custom logic induction depending on specific requirements needed by the parking monitor.

3 Concept description

Block Diagram:

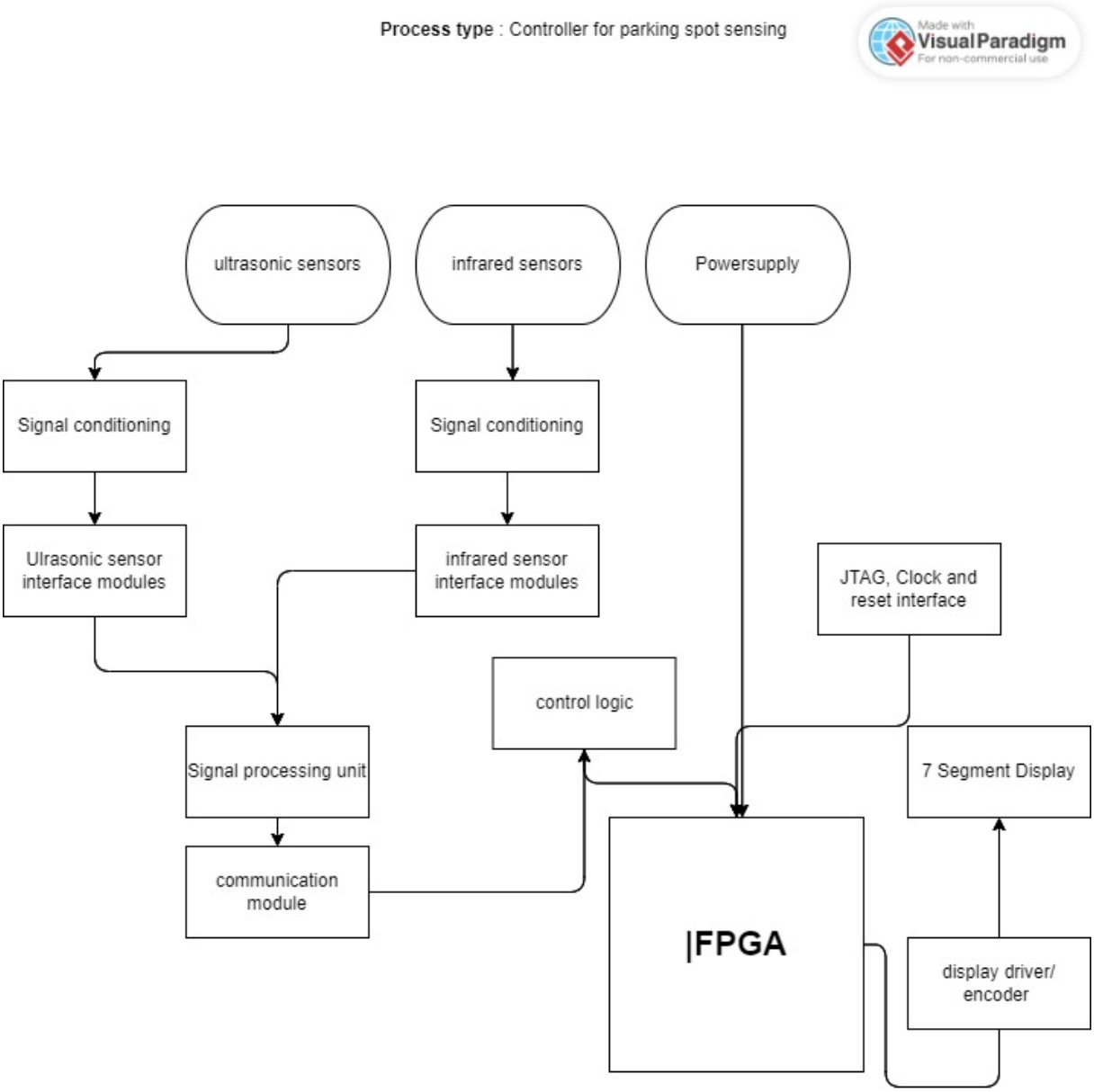


Fig (1). Block Diagram

Requirement diagram:

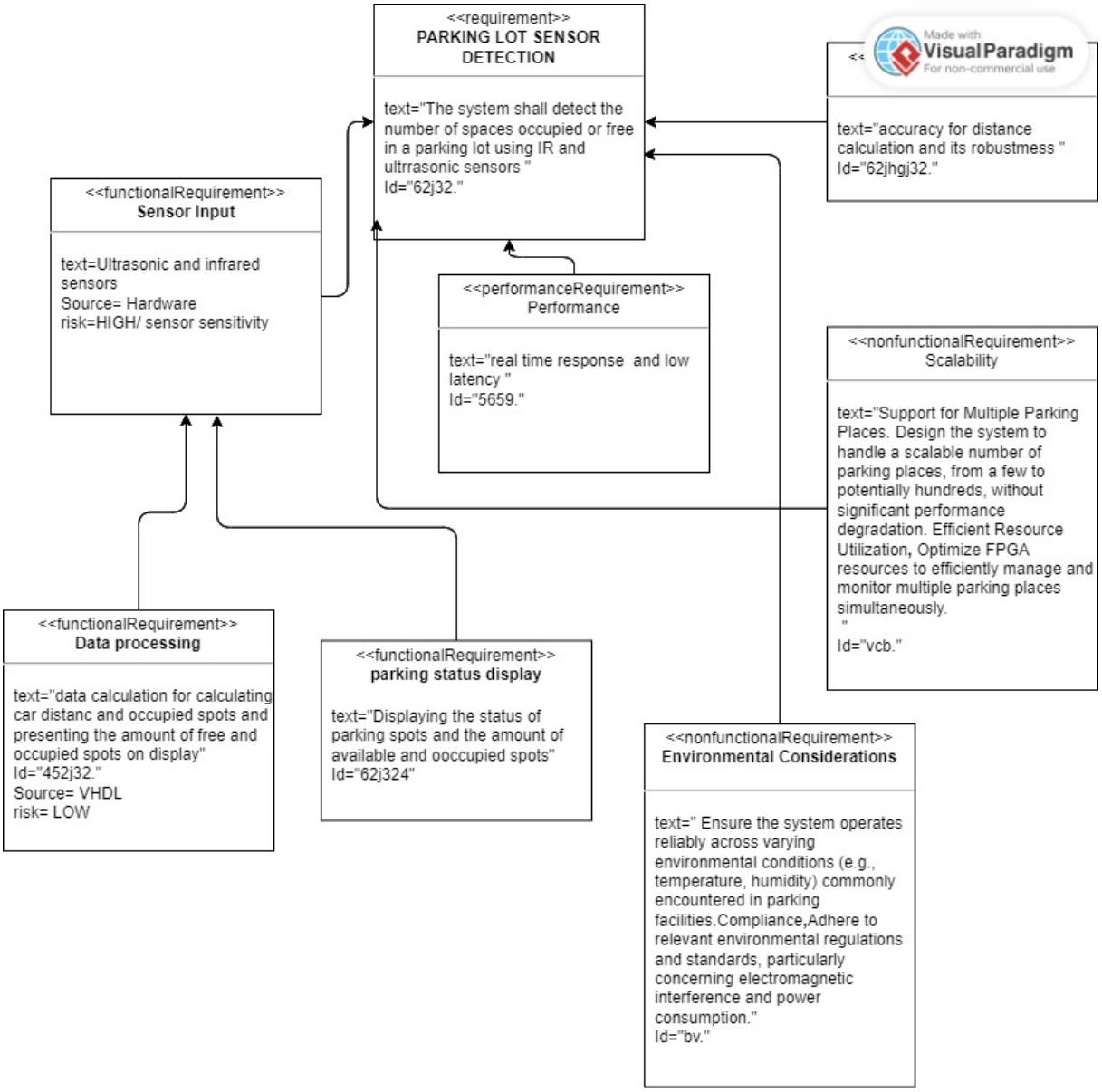


Fig (2). Requirement diagram

The system continuously surveys the availability of parking spaces on real-time basis by using ultrasonic and infrared sensors, this helps drivers to get immediate information about empty and filled parking spaces hence saving time for the searching of a suitable place to park. The displays are multiple 7 segments that show how many parking slots are filled and which specific slot is free so the driver can directly drive up to the slot without the need to drive through the whole parking lot to find the parking spot. Ultrasonic sensor implemented in this project capable of determining how close the edge of the car is to a line which marks the parking space. This system will make a buzzer sound in case there is any possibility of a collision. It can help prevent minor accidents as well as damages on vehicles or even property. Hence, all numbers of spots can be effectively monitored by it. It can be scaled up towards this goal. It can therefore work for huge car parks, multilevel garages, malls, airports and other places with a lot of traffic demanding efficient parking control. Any data collected through this

system enables facility manager perform analysis on parking patterns hence optimizing vehicle storage space use. As a result, the flow of traffic inside such premises is improved, and there is less congestion as well as effective utilization of available park resources. Implementation of our idea will not only provide easier solutions to complicated situations; it will also help reduce number of employees require to maintain a parking spot. Our idea will help create smart parking lots which will eliminate one of the biggest problems in the modern society.

4 Project/Team management

	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5
Ammar	Concept	Implementation	VHDL Code	Programing Fpga	PCB
Kashif	Concept	Implementation	VHDL Code	Programing Fpga	PCB
Hammad	Concept	Implementation	VHDL Code	Programing Fpga	PCB

.

5 Technologies

- *VHDL*
- *FPGA*
- *KiCAD*

6 VHDL and FPGA Implementation

Design Specification:
The project commenced with an in-depth analysis of the requirements and functionality expected from the parking management system. The design specification outlined the various components, their interactions, and the desired behavior of the system.

Architecture Design:
Based on the design specification, the architecture of the parking management system was defined. The different modules, such as sensor data processing, real-time status updating, and user notification, were identified and their interconnections established. The FPGA logic circuits were allocated and partitioned accordingly to accommodate the required functionalities.

VHDL Coding:
VHDL (VHSIC Hardware Description Language) was used to describe the behavior of the parking management system at the register-transfer level. Each module was implemented as a separate VHDL entity, defining its inputs, outputs, and internal logic. The behavior of each module was described using VHDL processes, signals, and concurrent statements.

RTL Simulation:
To ensure the correctness of the design, RTL (Register Transfer Level) simulation was performed. Test benches were created to generate stimuli and simulate the operation of the parking management system. The simulation results were analyzed to verify that the design met the desired specifications and produced the expected outputs.

Synthesis:
Once the RTL simulation was successful, the design was synthesized into a netlist using synthesis tools specific to the FPGA platform. The synthesis process mapped the VHDL code to FPGA resources, such as lookup tables, flip-flops, and interconnects. The synthesized netlist represented the low-level implementation of the design in terms of FPGA resources.

Place and Route:
The synthesized netlist was subjected to the place and route (P&R) process. P&R tools determined the physical locations of the design's components on the FPGA chip and established the interconnections between them. Constraints were applied to ensure proper timing, power optimization, and other design considerations.

Bitstream Generation:
Once the design was successfully placed and routed, the bitstream, a binary file containing configuration data for the FPGA, was generated. The bitstream file contained the programming information necessary to configure the FPGA with the implemented design.

FPGA Configuration:
The final step involved configuring the FPGA with the generated bitstream file. The bitstream was loaded onto the FPGA using programming tools specific to the FPGA platform. Once programmed, the FPGA executed the digital design, allowing the parking management system to function according to the specified behavior.

7 VHDL Code Implementation

The VHDL code for the parking management system was developed using a Finite State Machine (FSM) approach, ensuring efficient processing of sensor data and effective management of parking operations. The system operates in several states: idle, sensor check, update display, and warning. These states show the behavior of the parking system.

Finite State Machine Implementation

- **Idle State**: The system remains in this state until sensor input is detected.
- **Sensor Check State**: The system processes data from the IR and ultrasonic sensors.
    - **IR Sensor**: Determines if a parking spot is occupied or vacant.
    - **Ultrasonic Sensor**: Measures the distance to detect potential collisions.
- **Update Display State**: The status of each parking spot is updated on the 7-segment displays.
- **Warning State**: If a car is too close to the sensor, the piezo buzzer is activated.

IR Sensor and 7-Segment Display Code: The VHDL code for the IR sensor detects the presence of a vehicle in a parking spot. The sensor data is processed to update the status of the parking

spots, displayed on individual 7-segment displays. Each display shows '0' for available and '1' for occupied.

Ultrasonic Sensor and Piezo Buzzer Code: The ultrasonic sensor measures the distance between the car's edge and the parking spot border. If the distance falls below a predefined threshold, the system triggers a warning by sounding a piezo buzzer, alerting the driver of potential collisions.

Total Available Parking Spots Display: A separate 7-seg displays the total number of available parking spots, updated in real-time based on sensor data.

Code Snippet for IR Sensor and 7-Segment Display

```
process(clk)
begin
   if rising_edge(clk) then
      if ir_sensor_input = '1' then
         parking_status <= "1"; -- occupied
      else
         parking_status <= "0"; -- available
      end if;
      seven_segment_display <= parking_status;
   end if;
end process;
```

Code Snippet for Ultrasonic Sensor and Piezo Buzzer

```
process(clk)
begin
   if rising_edge(clk) then
      distance <= ultrasonic_sensor_reading;
      if distance < threshold then
         buzzer <= '1'; -- activate buzzer
      else
         buzzer <= '0'; -- deactivate buzzer
      end if;
   end if;
end process;
```

The VHDL implementation ensures real-time monitoring and management of parking spots, providing accurate status updates and collision warnings, thereby enhancing the overall efficiency and safety of the parking management system.

8 PCB Design

The PCB design for our parking management system was created using KiCad. The process began with developing the schematic, which involved selecting and connecting various components. We focused on understanding how each component should be interconnected to ensure correct functionality. After completing the schematic design, we assigned

appropriate footprints to the electronic components used in the project. Following this, an Electrical Rule Check (ERC) was conducted, and all errors were corrected, resulting in zero errors on the ERC.

Components and Schematic Design

Power Supply Circuit The power supply circuit provides stable voltage levels necessary for the operation of the entire system. It includes components like voltage regulators (LM317 and LM7805), capacitors, resistors, and an LED indicator for power status. The power supply ensures that the FPGA and other components receive the required 3.3V and 5V power.

FPGA Utilities

- **Clock (CLK) Circuit**: The oscillator circuit generates a stable clock signal required for the FPGA's operations. This ensures that the system runs at the correct frequency.
- **JTAG Interface**: The JTAG interface is used for programming and debugging the FPGA. It includes the necessary connections for TCK, TDI, TDO, TMS, and other control signals.

FPGA Externals

- **7-Segment Displays**: The 7-segment displays show the status of each parking spot (0 for available and 1 for occupied) and the total number of available spots. The displays are connected to the FPGA via GPIO pins, allowing real-time updates based on sensor inputs.
- **Processor with GPIO**: The processor handles input and output operations, processing data from the sensors and updating the display and buzzer. GPIO pins are used to interface with external components like sensors, displays, and buzzers.

PCB Layout and Footprints

The PCB layout was designed to be compact yet functional, keeping manufacturing costs low without compromising on performance. Small but effective footprints were assigned to each component, ensuring optimal use of space on the PCB. The layout ensures proper signal integrity and minimizes interference between different sections of the circuit.

Layout Considerations

- **Power Supply Lines**: Ensured to be robust and well-routed to provide stable power to all components.
- **Signal Traces**: Routed to minimize cross-talk and interference, especially between the clock signal and other sensitive traces.
- **Component Placement**: Organized to allow easy access for debugging and to maintain a clear flow of signals between related components.

Electrical Rule Check (ERC)

After completing the schematic and layout, an Electrical Rule Check (ERC) was performed to identify any potential design issues. All detected errors were corrected, ensuring a reliable

and error-free PCB design. The final design passed the ERC with zero errors, confirming its correctness.

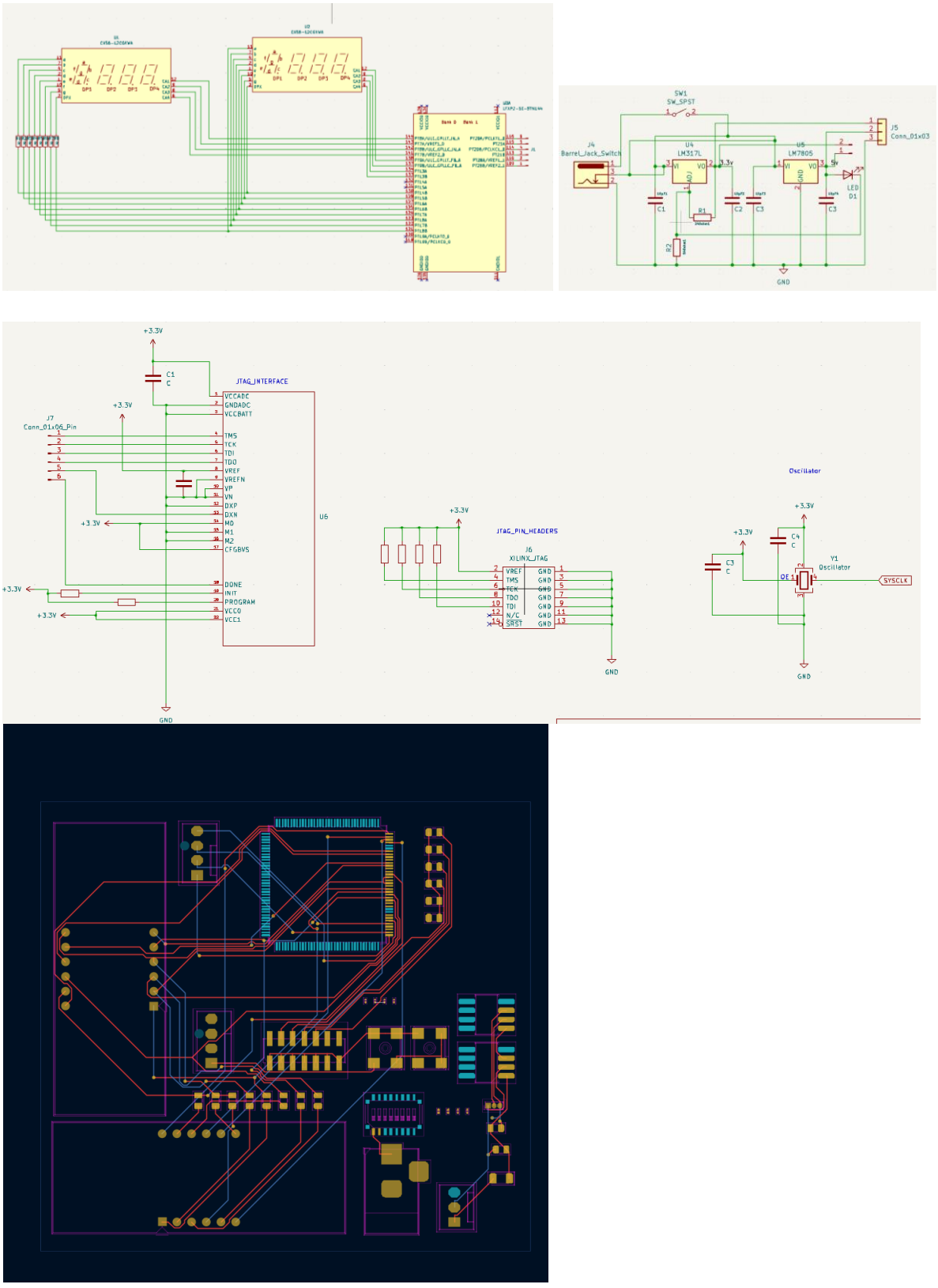
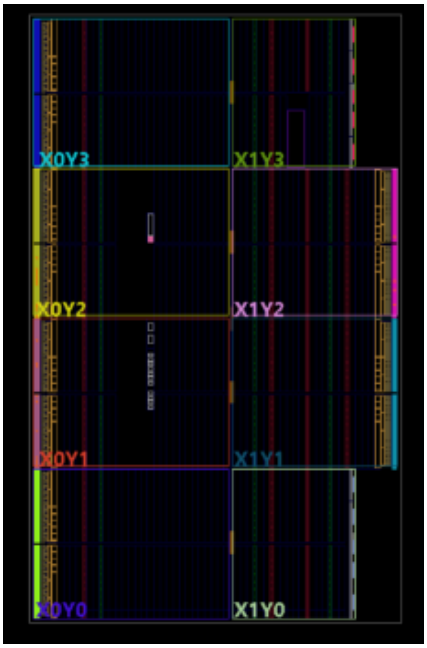


Fig (3). PCB schematics and routing

9 Synthesis Design and Implementation Design

The parking management system design was successfully implemented on the Artix 7 FPGA. The VHDL code was synthesized and mapped, resulting in a functional design. A finite state machine (FSM) was employed to control the system's behavior, with states including Idle, Sensor Check, Update Display, and Warning. The FSM transitions between states based on sensor inputs and real-time parking status. The figure below shows the implementation design generated on Xilinx Vivado software:



This design demonstrates the successful synthesis and implementation of the parking management system on the Artix 7 FPGA, showcasing the practical application of FPGA technology in real-world scenarios.

10 Sources/References

- Learning resources. KiCad EDA. (n.d.).
- https://www.kicad.org/help/learning-resources/
- Surf-VHDL. (2021, December 27). Vivado project tutorial - surf-VHDL. Surf. https://surf-vhdl.com/vivado-project-tutorial/
- FPGA Programming for Beginners Bring your ideas to life by creating hardware designs and electronic circuits with SystemVerilogb Frank Bruno

AFFIDAVIT

I hereby confirm that I have written this paper independently and have not used any sources or aids other than those indicated. All statements taken from other sources in wording or sense are clearly marked. Furthermore, I assure that this paper has not been part of a course or examination in the same or a similar version.

NAME	Ammar Imran Khan	Kashif Raza	Hammad Karamat
LOCATION	Lippstadt	Lippstadt	Lippstadt
DATE	28/06/2024	28/06/2024	28/06/2024
SIGNATIRE	X Ammar	X Kashif	X Hammad