



FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER AND COMMUNICATION SYSTEMS ENGINEERING

DIGITAL SYSTEMS

EEE3822-1

GROUP PROJECT REPORT

SEMESTER II 2022/2023

PERKUP ENTERPRISE: REWARDIFY

LECTURER: PN. ROSLIZAH BINTI ALI

GROUP 7 MEMBERS:

| NO. | NAME | MATRIC NO. |
|-----|------------------------------------|------------|
| 1. | MUHAMMAD IRFAN BIN ABDUL AZEEZ | 211226 |
| 2. | TASNIM MAHDIYA | 209488 |
| 3. | SITI AFIQAH HUSNA BINTI AMIRRUDDIN | 210862 |
| 4. | MUHAMMAD FARIS IRFAN BIN YAAKOB | 210185 |

TABLE OF CONTENTS

| | |
|-----------------------------|----|
| PROJECT INTRODUCTION | 3 |
| PROJECT MANAGEMENT | 6 |
| DESIGN METHODOLOGY | 9 |
| DESIGN DESCRIPTION | 10 |
| DESIGN ANALYSIS AND RESULTS | 16 |
| CONCLUSIONS | 23 |
| REFERENCES | 26 |
| APPENDIX | 26 |

1.0 PROJECT INTRODUCTION

1.1 Background and Problem Statement

The absence of a rewards system in the working environment nowadays makes it challenging for organizations to effectively motivate and celebrate employees, which reduces employees' engagements and results in poor performance. Reward systems implemented by organizations will influence employees' behavior and attitude towards their job if the rewards satisfy their needs and help them to succeed in their personal goals [1]. Thus, our company aims to introduce a card-based points rewards system that will boost employee motivation and performance by offering a comprehensive rewards program. Employees will earn points based on their achievements and positive contributions, which can be redeemed for various incentives. However, the unique aspect of our system is that the card will also track employees' ethical behavior, allowing for the deduction of points when misconduct occurs. The card's social score will determine access to certain facilities, creating an environment that promotes ethical conduct and accountability. The proposed project aims to design a smart card system using a Verilog code FPGA DE1 SoC board that can be developed to achieve sustained, inclusive and sustainable economic growth, productive employment and decent work. This project will align to UN Sustainable Development Goals (SDGs) 8 which applies to Decent Work and Economic Growth.

1.2 Project Objectives

1. Improve employee efficiency policies.
2. Help companies build a winning work culture.
3. Promote employee development oriented policies
4. Set the trendlines of today's smart card system of organization.

1.3 Design Specifications

| Sensors | Input/Output | Remarks |
|-------------------|--------------|--|
| IR Sensors | Input | To detect the card. |
| LED | Output | To show that the employee's ID exists in the system's corporate database. |
| DC Geared Motor | | To indicate that the employees will have access to the certain company's facilities. |
| 7-Segment Display | | To display the employees' social score and reward points. |
| Buzzer | | To indicate that the employee is not from the company. |

1.4 Feature Set

Rewardify is designed as a robust platform that enables employees to accumulate reward points depending on their performance and behavior inside the company's organization [2]. A **social score feature** that is given by the company (e.g. the project manager) will also be included in the system to give a comprehensive overview of a worker's behavior and contributions. Through our product, employees will also be able to exchange their earned **reward points** for several kinds of incentives and presents. The target audience mainly consists of large corporations and medium-sized companies.

1.5 Project Scope

1..5.1 Deliverables:

- A functional prototype system integrating the DE1-SoC board and necessary components for access control.
- Implementation of an access control system based on employee social scores.

- Accurate verification of employee IDs and control of facility access.
- Visual and audio feedback indicating access granted or restricted.
- User-friendly interface displaying employee credentials and transaction details.

1.5.2 Exclusions

- Keypad and LCD Display as the components that we have excluded in our projects due to better input and output choices.
- Advanced features that are not essential to the projects.

1.5.3 Constraints

- Resource Constraint

We obtained our IR sensor a little late and close to the deadline. This caused us to rush our finishing touches.

- Time Constraint

Important deadlines given are taken into account. All of the deliverables and milestones are monitored to guarantee timely completion.

- Cost Constraint

The purchase cost of electronic components like sensors should be considered. Choosing components that satisfy the project requirements and are affordable might help keep spending under control.

1.5.4 Assumptions

- It is expected that a reliable and sufficient power source is provided to the FPGA board and sensors. Within the defined operating conditions, power requirements and any power-related problems, such as voltage levels or current restrictions, are taken into account.
- It is expected that the selected sensors and FPGA board can be successfully interfaced with one another and are compatible.

2.0 PROJECT MANAGEMENT

2.1 Project Risk

1. Circuit connection did not function as envisioned

We use various different components ranging from IR sensor to buzzer and motor. Therefore, each of the component needs to be in perfect working condition for us to succeed in this project

2. Code did not function as intended

The coding part can be very tricky. Hence, an efficient and concise code needs to be written to ensure our project works well.

3. Component breaks down

This is a fear factor for us as we have burned through multiple components that fried up. An accurate amount of voltage needs to flow in order to avoid any component being damaged.

2.3 Project Costing

| Component | Quantity | Price (RM) |
|--------------------|----------|------------|
| IR Obstacle Sensor | 2 | 10.00 |
| Jumper Cable | 2 | 4.00 |
| Keypad (4x3) | 1 | 15.00 |
| DC Geared Metal | 1 | 6.00 |
| 9V Battery | 1 | 15.80 |
| 2x16 LCD Display | 1 | 10.00 |
| Total | | 60.8 |

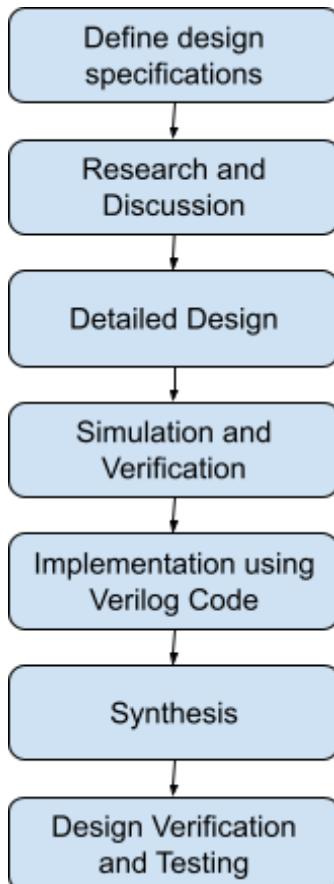
2.4 Project Gantt Chart

3.0 DESIGN METHODOLOGY

3.1 Design approach

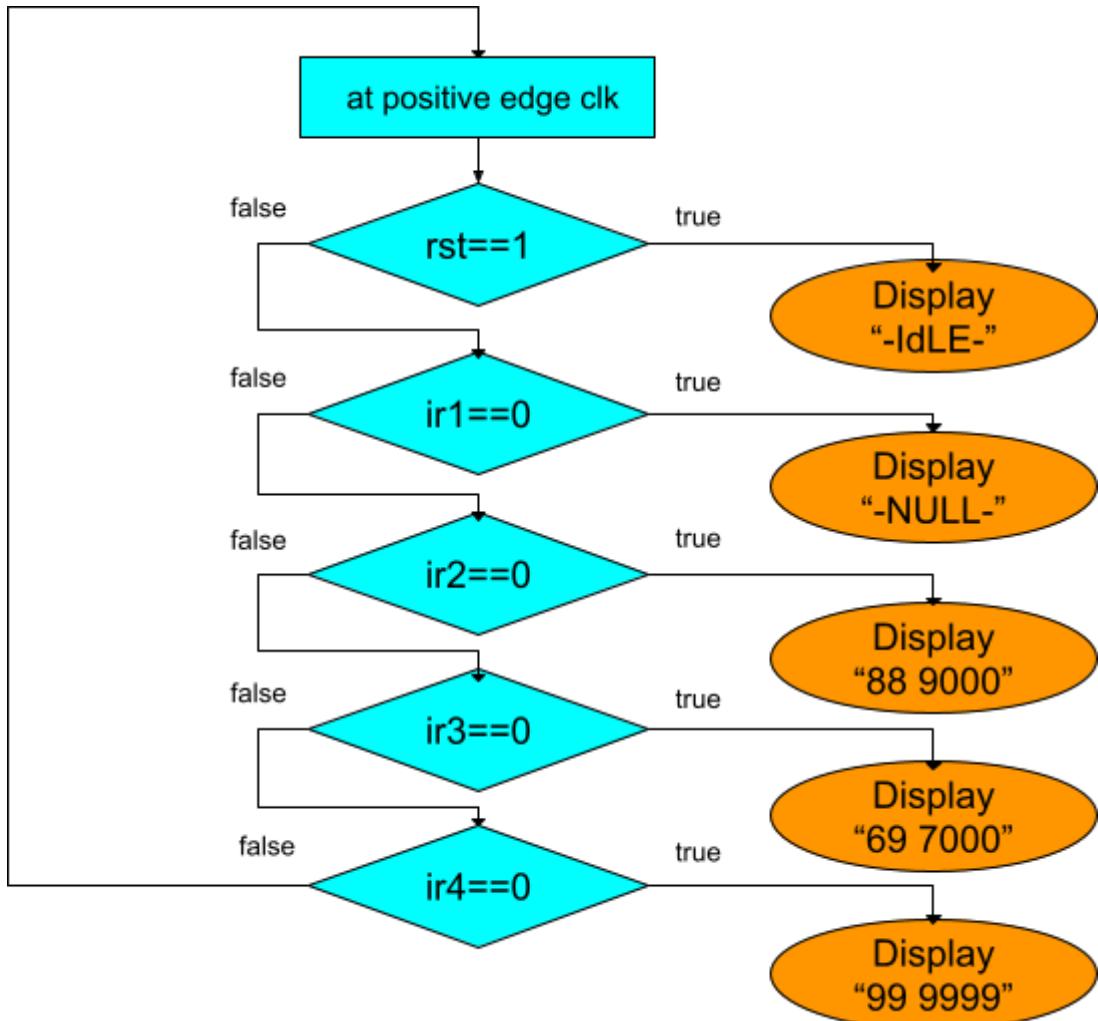
In our project, we started off by defining the design specification and goals. The project's objectives, goals, and desired results should all be clearly stated. Other than that, the specific requirements and limitations are also identified related to the digital design, including functionality, performance, cost, and schedule. We have also got a thorough understanding of the project requirements by gathering information from responders and additional research to similar existing designs and technologies. Various design strategies and methods that are relevant to our project have also been explored. Based on the project requirements, early design concepts through brainstorming with the group members have also been decided and discussed so that all of the group members have one full understanding of an overall project. To find and fix any design errors, functional simulations are run and the outcomes are analyzed. Then, programming is done through Verilog and ends with design verification and testing.

Design flow is shown below:



4.0 DESIGN DESCRIPTION

4.1 ASM charts



4.2 Description of top-level design

```
module RewardifySystem(
    input wire clk, rst, ir1, ir2, ir3, ir4,
    output reg led1, led2, buz, mot,
    output wire [6:0] hex0, hex1, hex2, hex3, hex4, hex5
);

    wire [6:0] wire_hex0, wire_hex1, wire_hex2, wire_hex3, wire_hex4, wire_hex5;

    assign hex0 = wire_hex0;
    assign hex1 = wire_hex1;
    assign hex2 = wire_hex2;
    assign hex3 = wire_hex3;
```

```

assign hex4 = wire_hex4;
assign hex5 = wire_hex5;

displaySystem display(
    .ir1(ir1), .ir2(ir2), .ir3(ir3), .ir4(ir4),
    .clk(clk), .rst(rst),
    .hex0(wire_hex0), .hex1(wire_hex1), .hex2(wire_hex2),
    .hex3(wire_hex3), .hex4(wire_hex4), .hex5(wire_hex5)
);

always @(posedge clk)
begin
    if (rst)                                //if reset is triggered
        begin
            led1 <= 1'b0;
            led2 <= 1'b0;
            buz <= 1'b0;
            mot <= 1'b1;
        end
    else if (ir1 == 1'b0)                      //if user4 is triggered
        begin
            led1 <= 1'b1;
            led2 <= 1'b0;
            buz <= 1'b1;
            mot <= 1'b1;
        end
    else if (ir2 == 1'b0)                      //if user1 is triggered
        begin
            led2 <= 1'b1;
            led1 <= 1'b0;
            buz <= 1'b0;
            mot <= 1'b0;
        end
    else if (ir3 == 1'b0)                      //if user2 is triggered
        begin
            led2 <= 1'b1;
            led1 <= 1'b0;
            buz <= 1'b0;
            mot <= 1'b1;
        end
    else if (ir4 == 1'b0)                      //if user3 is triggered
        begin
            led2 <= 1'b1;
            led1 <= 1'b0;
            buz <= 1'b0;
            mot <= 1'b0;
        end
    end
endmodule

```

On the top level module, several variables have been defined to help implement the Rewardify core system. The input variables are clk, rst, ir1, ir2, ir3 and ir4 whereas output variables are led1, led2, buz, mot, hex0, hex1, hex2, hex3, hex4 and hex5.

The variable assigned are as follow:

- clk → assigned to board's 50 Mhz clock frequency
- rst → assigned to board's slide switch
- ir1 → assigned to external I/O (IR obstacle sensor to detect non existing user)
- ir2 → assigned to external I/O (IR obstacle sensor to detect user 1)
- ir3 → assigned to external I/O (IR obstacle sensor to detect user 2)
- ir4 → assigned to external I/O (IR obstacle sensor to detect user 3)
- led1 → assigned to external I/O (red LED)
- led2 → assigned to external I/O (green LED)
- buz → assigned to external I/O (buzzer)
- mot → assigned to external I/O (DC motor)
- hex0 - hex5 → assigned to board's six 7-segment displays

Firstly, at every posedge of *clk*, the values of *rst*, *ir1*, *ir2*, *ir3* and *ir4* will be read continuously inside the 'always' block. *ir1*, *ir2*, *ir3* and *ir4* are **active low**; hence, the condition inside the related 'if else' block is always comparing it to '0'. Other than that, *mot* and *hex0-hex5* are also **active low**. In each of the satisfied conditions of 'if else' cases, it will assign specified values to each output variable above. As example, if *ir1* detects any obstacle(finger) within certain distance ranges, it will send value 0 to the board to compare the values in the 'always' block, here the *else if (ir1 == 1'b0)* code snippet will be executed since it satisfies the condition. The output that will be observed by the user is the buzzer and red LED will be turned on until a new value has changed inside the 'always' block sensitivity list.

4.3 Description of sub-module

```
module displaySystem (
    input ir1, ir2, ir3, ir4, clk, rst,
    output reg [6:0] hex0, hex1, hex2, hex3, hex4, hex5
);

    always @(posedge clk)
    begin
        if (rst)
            begin
                hex0 = 7'b0111111;
                hex1 = 7'b0000110;
                hex2 = 7'b1000111;
                hex3 = 7'b0100001;
                hex4 = 7'b1111001;
                hex5 = 7'b0111111; // display -IdLE-
            end
        end
        else if (ir1 == 1'b0)
        begin
            hex0 = 7'b1111111;
```

```

hex1 = 7'b1111111;
hex2 = 7'b0001001;
hex3 = 7'b1000001;
hex4 = 7'b0001001;
hex5 = 7'b1111111; // display -NULL-
end
else if (ir2 == 1'b0)
begin
    hex0 = 7'b1000000;
    hex1 = 7'b1000000;
    hex2 = 7'b1000000;
    hex3 = 7'b0011000;
    hex4 = 7'b0000000;
    hex5 = 7'b0000000; // display 88 9000
end
else if (ir3 == 1'b0)
begin
    hex0 = 7'b1000000;
    hex1 = 7'b1000000;
    hex2 = 7'b0110000;
    hex3 = 7'b1111000;
    hex4 = 7'b0011000;
    hex5 = 7'b0000010; // display 69 7300
end
else if (ir4 == 1'b0)
begin
    hex0 = 7'b0011000;
    hex1 = 7'b0011000;
    hex2 = 7'b0011000;
    hex3 = 7'b0011000;
    hex4 = 7'b0011000;
    hex5 = 7'b0011000; // display 99 9999
end
end
endmodule

```

Referring back to the top level module, a **sub-module instantiation by order** is specified in the code to pass some variables values to the displaySystem module. This module is used to display the social credit score as well as reward points accumulated by the user id on the six 7-segment displays (HEX0,HEX1,HEX2,HEX3,HEX4 and HEX5) on the board. Continuing from the test example in the top module section just now, the six 7-segment display will also display “-NULL-” alongside the working red LED and buzzer. The display on the 7-segment displays changes from time to time depending on the changes on ir sensors.

A single module code that works just the same as two combined module above:

```
module RewardifySys(
    input wire clk,rst,ir1,ir2,ir3,ir4,
    output reg led1,led2,buz,mot,
    output reg [6:0]hex0,hex1,hex2,hex3,hex4,hex5);

    //led1 is red, led2 is green

    //ir1 is for red led, ir2 is for green led

    always@(posedge clk)
    begin
        if(rst)
        begin
            led1<=1'b0;
            led2<=1'b0;
            buz<=1'b0;
            mot<=1'b1;
            hex0=7'b0111111;
            hex1=7'b0000110;
            hex2=7'b1000111;
            hex3=7'b0100001;
            hex4=7'b1111001;
            hex5=7'b0111111;                                //display -IdLE-
        end
        else if(ir1==1'b0)
        begin
            led1<=1'b1;
            led2<=1'b0;
            buz<=1'b1;
            mot<=1'b1;
            hex0=7'b1111111;
            hex1=7'b1111111;
            hex2=7'b0001001;
            hex3=7'b1000001;
            hex4=7'b0001001;
            hex5=7'b1111111;                                //display -NULL-
        end
        else if(ir2==1'b0)
        begin
            led2<=1'b1;
            led1<=1'b0;
            buz<=1'b0;
            mot<=1'b0;
            hex0=7'b1000000;
            hex1=7'b1000000;
            hex2=7'b1000000;
            hex3=7'b0011000;
            hex4=7'b0000000;
            hex5=7'b0000000;                                //display 88 9000
        end
        else if(ir3==1'b0)
        begin
```

```

led2<=1'b1;
led1<=1'b0;
buz<=1'b0;
mot<=1'b1;
hex0=7'b1000000;
hex1=7'b1000000;
hex2=7'b0110000;
hex3=7'b1111000;
hex4=7'b0011000;
hex5=7'b0000010;                                //display 69 7300
end
else if(ir4==1'b0)
begin
led2<=1'b1;
led1<=1'b0;
buz<=1'b0;
mot<=1'b0;
hex0=7'b0011000;
hex1=7'b0011000;
hex2=7'b0011000;
hex3=7'b0011000;
hex4=7'b0011000;
hex5=7'b0011000;                                //display 99 9999
end
end
endmodule

```

Though it is simple and easy to understand the flow of code if all of the variables are instantiated within a single module, It is a good practice to do hierarchical design methods when we are doing much bigger and complex designs.

5.0 DESIGN ANALYSIS AND RESULTS

5.1 Simulation and pictorial results

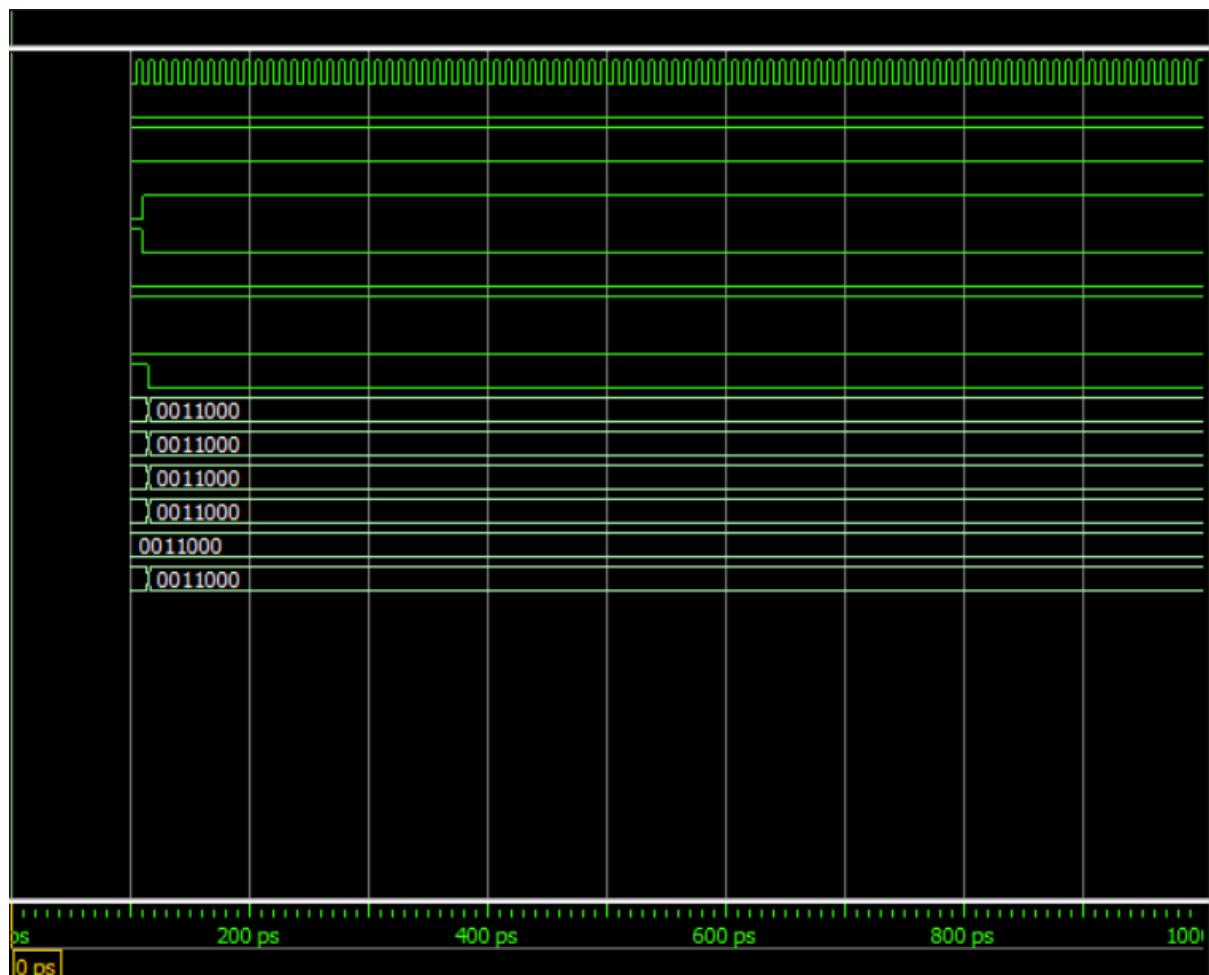


Figure 1.1: Wave form of the test bench

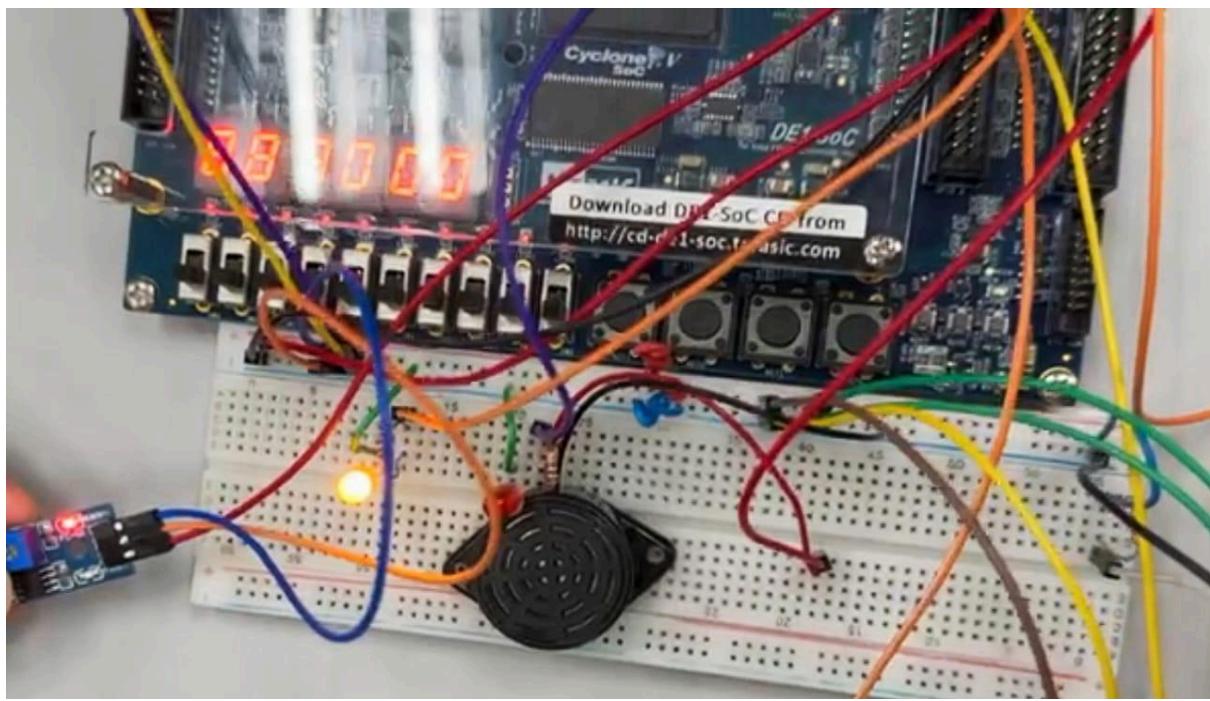


Figure 1.2: Pictorial result of user 1

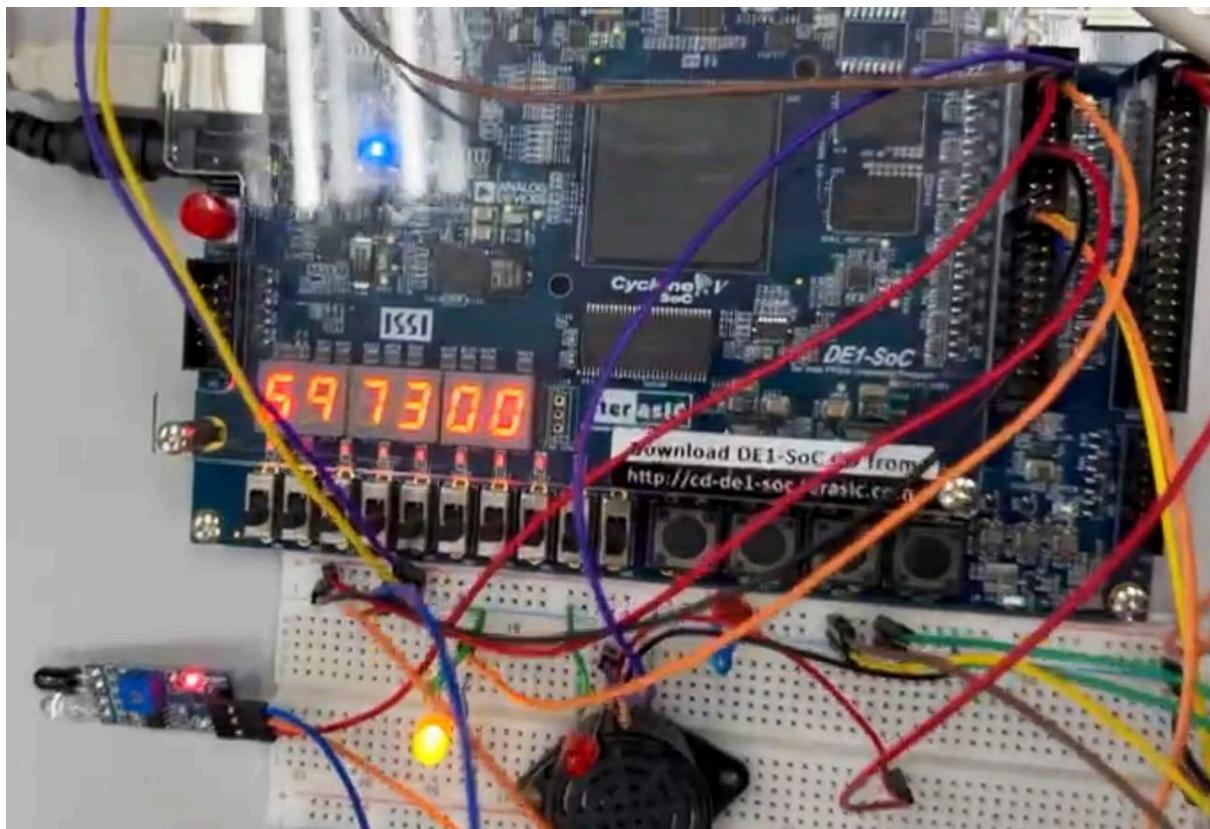


Figure 1.3: Pictorial result of user 2

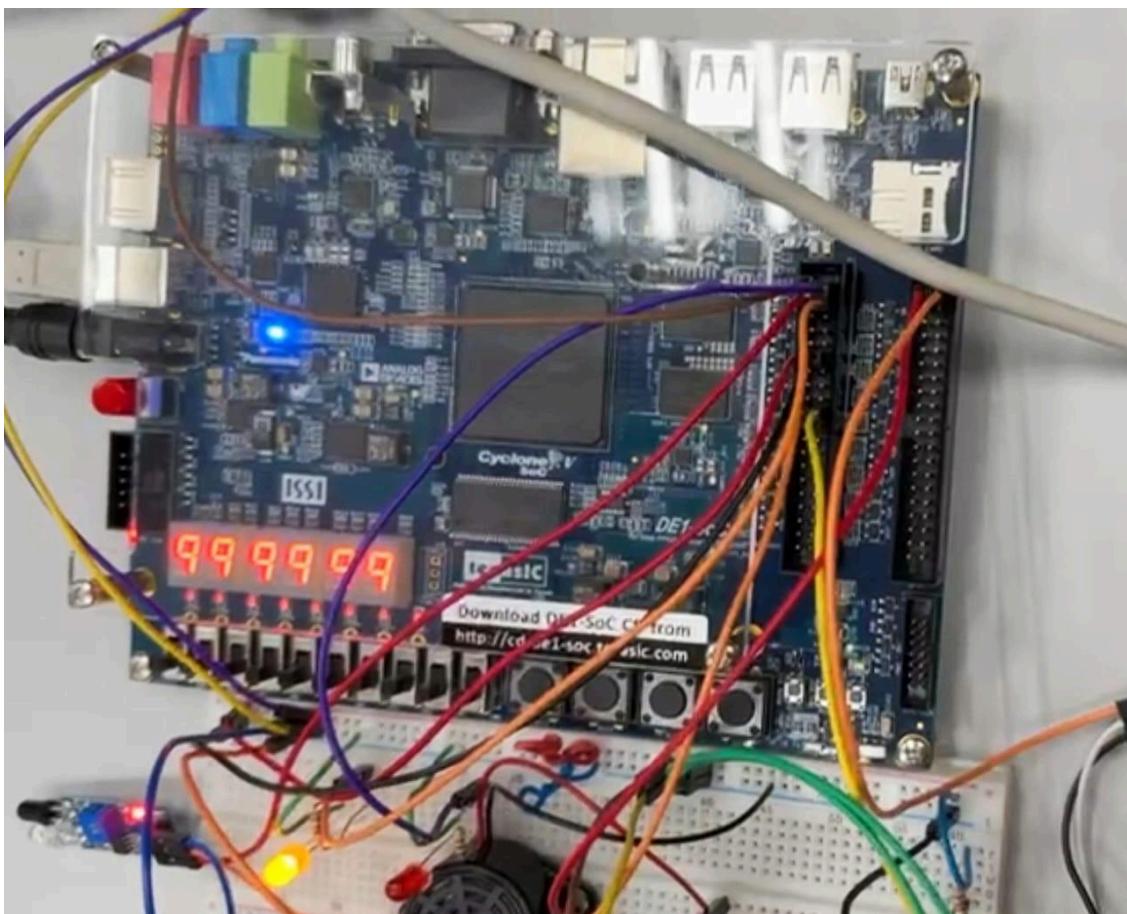


Figure 1.4: Pictorial result of user 3

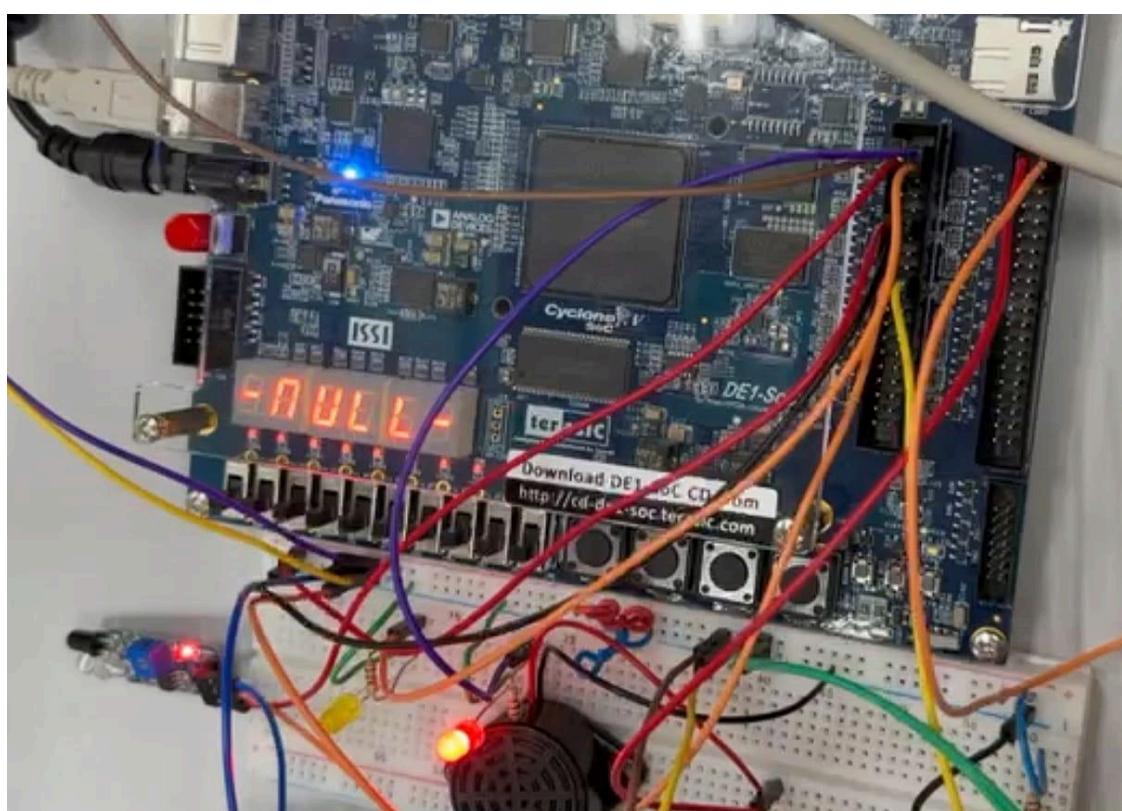


Figure 1.4: Pictorial result of user 4

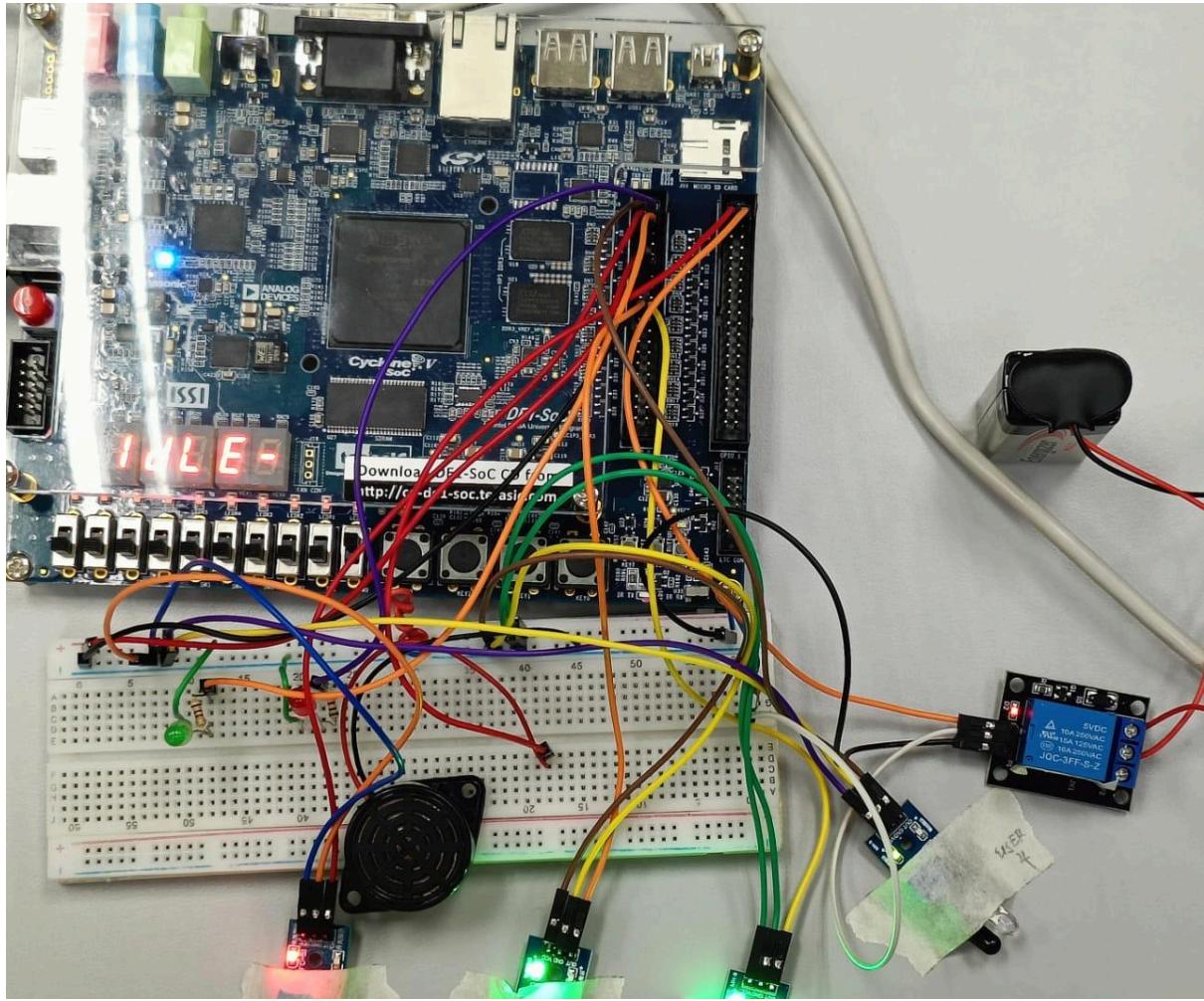


Figure 1.5: Pictorial result of reset

5.2 Implementation Statistics

5.2.1 Resource usage

| Analysis & Synthesis Resource Usage Summary | | |
|---|---|-------|
| <<Filter>> | | |
| | Resource | Usage |
| 1 | Estimate of Logic utilization (ALMs needed) | 9 |
| 2 | | |
| 3 | Combinational ALUT usage for logic | 14 |
| 1 | -- 7 input functions | 0 |
| 2 | -- 6 input functions | 0 |
| 3 | -- 5 input functions | 1 |
| 4 | -- 4 input functions | 0 |
| 5 | -- <=3 input functions | 13 |
| 4 | | |
| 5 | Dedicated logic registers | 15 |
| 6 | | |
| 7 | I/O pins | 52 |
| 8 | | |
| 9 | Total DSP Blocks | 0 |
| 10 | | |
| 11 | Maximum fan-out node | buz~1 |
| 12 | Maximum fan-out | 15 |
| 13 | Total fan-out | 184 |
| 14 | Average fan-out | 1.38 |

Figure 2.1: Resource usage

| Analysis & Synthesis Resource Utilization by Entity | | | | | | | | | |
|---|---------------------|---------------------------|-------------------|------------|------|--------------|---------------------|-------------|--------------|
| <<Filter>> | | | | | | | | | |
| litation Hierarchy Node | Combinational ALUTs | Dedicated Logic Registers | Block Memory Bits | DSP Blocks | Pins | Virtual Pins | Full Hierarchy Name | Entity Name | Library Name |
| | 14 (14) | 15 (15) | 0 | 0 | 52 | 0 | jirfan | jirfan | work |

Figure 2.2: Resource utilization

5.2.2 Timing statistics

| Timing Analyzer Summary | |
|-------------------------|--|
| <<Filter>> | |
| Quartus Prime Version | Version 22.1std.1 Build 917 02/14/2023 SC Lite Edition |
| Timing Analyzer | Legacy Timing Analyzer |
| Revision Name | irfan |
| Device Family | Cyclone V |
| Device Name | 5CSEMA5F31C6 |
| Timing Models | Final |
| Delay Model | Combined |
| Rise/Fall Delays | Enabled |

Figure 3.1: Timing analyzer

| Parallel Compilation | | |
|----------------------|----------------------------|-------------|
| <<Filter>> | | |
| | Processors | Number |
| 1 | Number detected on machine | 12 |
| 2 | Maximum allowed | 12 |
| 3 | | |
| 4 | Average used | 1.09 |
| 5 | Maximum used | 12 |
| 6 | | |
| 7 | ▼ Usage by Processor | % Time Used |
| 1 | Processor 1 | 100.0% |
| 2 | Processor 2 | 1.7% |
| 3 | Processor 3 | 1.7% |
| 4 | Processors 4-12 | 0.6% |

Figure 3.2: Parallel compilation of time analyzer

| Clocks | | | | | | | | | | | | | | | | | |
|--------|------------|------|--------|------------|-------|-------|------------|-----------|-------------|-------|--------|-----------|------------|----------|--------|--------|---------|
| | <<Filter>> | | | | | | | | | | | | | | | | |
| | Clock Name | Type | Period | Frequency | Rise | Fall | Duty Cycle | Divide by | Multiply by | Phase | Offset | Edge List | Edge Shift | Inverted | Master | Source | Target |
| 1 | clk | Base | 1.000 | 1000.0 MHz | 0.000 | 0.500 | | | | | | | | | | | { clk } |

Figure 3.3: Clock period

5.3 Discussion of results

The condition for the system works if the user has reward points equal or more than 70. Otherwise the user will not be able to enter certain facilities. Following the condition, when the user 1 with 88 reward points is triggered as input, the system will turn on green light, the 7 segment display will show the social score and the reward points and the motor will start working. When the user 2 with 69 reward points is triggered as input, the system will turn on green light as the system can recognize the user registered in the system, but the motor will not work as the reward points are less than 70 which doesn't fulfill the condition. Similarly, The user 3 with 99 reward points will function as user 1 as it follows the system conditions. On the contrary, when user 4 tries to access the system, the red light will turn on as the system can not find the user 4 in the system registered. In that case, the buzzer will ring and the motor will not work. Moreover, the display will show NULL. To input again, the user needs to press the reset button , displaying IDLE and try again.

6.0 CONCLUSIONS

In summary, our digital design project has been successful in accomplishing the stated objectives and requirements in designing a smart card system that fulfills SDG 8 using the FPGA Board. With Verilog, we were able to create an FPGA-based solution that fulfills the requirements with at least three input and output. (buzzer, IR sensor and DC Geared Motor). The implementation of an access control system based on employee social scores, accurate verification of employee IDs and control of facility access and visual and audio feedback indicating access granted or restricted have been achieved. With buzzer, we have shown that the employees with no user ID in the respective company's system database will not have access to the facilities at all. Then, the DC Geared Motor has also shown their purpose by showing the employees' access to certain facilities. Other than that, we have also used 7-Segment Display to display the social score and rewards points. Furthermore, an IR sensor will be triggered once it detects hand motion and LED to indicate that the user ID card has been verified to be the company's own employee.

During our project, we also encountered several problems regarding the DC Geared Motor where one of the parts became damaged. Moreover, the DC Geared Motor also couldn't be run at first due to excessive power consumption. For that reason, we learned about how FPGAs have limited power consumption capabilities that depend on several factors. Hence, these potential factors lead to increasing the battery capacity and incorporating a DC delay relay so that the DC Geared Motor could be run successfully. Other than that, our group intended to use a keypad to demonstrate how the employees can enter their respective user ID and LCD Display to prove the verification of their ID. However, we discovered that the use of IR sensors is better and more suitable to illustrate a function of a card. Moreover, we encountered several coding errors in Verilog leading to failed desired results. The compilation of the Verilog code is time consuming. As a result, continuous failure in bugging and testing the code made it a waste of time. However, multiple strategies are done such as minimizing design complexity and improving the design for compilation efficiency so that the project is completed on time.

6.1 Independent Reflection

| | |
|------------------------------------|--|
| Muhammad Irfan Bin Abdul Azeez | I learned a lot about specific digital systems concepts and techniques. For example, I learned about the different types of digital logic gates, how to design a circuit, and how to program a FPGA board using different software development tools. Throughout the project, team members gave their all to ensure the completion of this project. In my opinion, I truly enjoyed doing this project as it helps us to apply the knowledge we have been gaining in the class. |
| Muhammad Faris Irfan Bin Yaakob | I now have a deep understanding on the working of an FPGA, a circuit and its various components and learned to implement a code into a circuit. This assignment can be improved by allowing a few standout groups to get their project recognised by companies. |
| Siti Afiqah Husna Binti Amirruddin | Through this subject, I have learned about Verilog, FPGA, and designing complex systems in detail. I believe that working in a group enhances problem-solving, communication between the group members and teamwork skills for efficient project completion. I have also learned a lot from other group members thus expanding my knowledge and understanding. |
| Tasnim Mahdiya | I acquired the knowledge of FPGA board hardware and software functionalities. Moreover, I was able to learn various real life applications of digital system design |

with different hardware components throughout the project. The project was a success on its own though there's always room for improvement; in that case, the project can be further improved by adding some features for example, the system can have a keypad system where a user can type their access number if by any chance they lose their card. Overall, the team project strengthened critical thinking, problem solving, communication and teamwork skills.

REFERENCES

- [1] F. Francis, C. T. Zirra, and M. I. Charles J., "Reward system as a strategy to enhance employees performance in an organization," *Archives of Business Research*, vol. 8, no. 6, pp. 156–164, 2020. doi:10.14738/abr.86.8403
- [2] Dr. H. Ghorbani, Dr. S. M. Abdollahi, and I. N. Mondanipour, "An Empirical Study on the Impacts of Market Orientation and Innovation on New Product Success (Case Study: Food Manufacturers in Isfahan, Iran)," *International Journal of Academic Research in Business and Social Sciences*, vol. 3, no. 9, Sep. 2013, doi: <https://doi.org/10.6007/ijarbss/v3-i9/214>.

APPENDIX

Top module:

```
module RewardifySystem(
    input wire clk, rst, ir1, ir2, ir3, ir4,
    output reg led1, led2, buz, mot,
    output wire [6:0] hex0, hex1, hex2, hex3, hex4, hex5
);

    wire [6:0] wire_hex0, wire_hex1, wire_hex2, wire_hex3, wire_hex4, wire_hex5;

    assign hex0 = wire_hex0;
    assign hex1 = wire_hex1;
    assign hex2 = wire_hex2;
    assign hex3 = wire_hex3;
    assign hex4 = wire_hex4;
    assign hex5 = wire_hex5;

    displaySystem display(
        .ir1(ir1), .ir2(ir2), .ir3(ir3), .ir4(ir4),
        .clk(clk), .rst(rst),
        .hex0(wire_hex0), .hex1(wire_hex1), .hex2(wire_hex2),
        .hex3(wire_hex3), .hex4(wire_hex4), .hex5(wire_hex5)
    );

    always @(posedge clk)
    begin
        if (rst)
            begin
                led1 <= 1'b0;
                led2 <= 1'b0;
                buz <= 1'b0;
                mot <= 1'b1;
            end
        else if (ir1 == 1'b0)
            begin
```

```

led1 <= 1'b1;
led2 <= 1'b0;
buz <= 1'b1;
mot <= 1'b1;
end
else if (ir2 == 1'b0)
begin
    led2 <= 1'b1;
    led1 <= 1'b0;
    buz <= 1'b0;
    mot <= 1'b0;
end
else if (ir3 == 1'b0)
begin
    led2 <= 1'b1;
    led1 <= 1'b0;
    buz <= 1'b0;
    mot <= 1'b1;
end
else if (ir4 == 1'b0)
begin
    led2 <= 1'b1;
    led1 <= 1'b0;
    buz <= 1'b0;
    mot <= 1'b0;
end
end
endmodule

```

Sub-module:

```

module displaySystem (
    input ir1, ir2, ir3, ir4, clk, rst,
    output reg [6:0] hex0, hex1, hex2, hex3, hex4, hex5
);

    always @(posedge clk)
    begin
        if (rst)
        begin
            hex0 = 7'b0111111;
            hex1 = 7'b00000110;
            hex2 = 7'b10000111;
            hex3 = 7'b0100001;
            hex4 = 7'b1111001;
            hex5 = 7'b0111111; // display -ldLE-
        end
    end

```

```

else if (ir1 == 1'b0)
begin
    hex0 = 7'b1111111;
    hex1 = 7'b1111111;
    hex2 = 7'b0001001;
    hex3 = 7'b1000001;
    hex4 = 7'b0001001;
    hex5 = 7'b1111111;                                // display -NULL-
end
else if (ir2 == 1'b0)
begin
    hex0 = 7'b1000000;
    hex1 = 7'b1000000;
    hex2 = 7'b1000000;
    hex3 = 7'b0011000;
    hex4 = 7'b0000000;
    hex5 = 7'b0000000;                                // display 88 9000
end
else if (ir3 == 1'b0)
begin
    hex0 = 7'b1000000;
    hex1 = 7'b1000000;
    hex2 = 7'b0110000;
    hex3 = 7'b1111000;
    hex4 = 7'b0011000;
    hex5 = 7'b0000010;                                // display 69 7300
end
else if (ir4 == 1'b0)
begin
    hex0 = 7'b0011000;
    hex1 = 7'b0011000;
    hex2 = 7'b0011000;
    hex3 = 7'b0011000;
    hex4 = 7'b0011000;
    hex5 = 7'b0011000;                                // display 99 9999
end
end
endmodule

```

Verilog code with only one functional module of the project:

```

module RewardifySys(
input wire clk,rst,ir1,ir2,ir3,ir4,
output reg led1,led2,buz,mot,
output reg [6:0]hex0,hex1,hex2,hex3,hex4,hex5);

//led1 is red, led2 is green

//ir1 is for red led, ir2 is for green led

always@(posedge clk)
begin
    if(rst)

```

```

begin
led1<=1'b0;
led2<=1'b0;
buz<=1'b0;
mot<=1'b1;
hex0=7'b0111111;
hex1=7'b0000110;
hex2=7'b1000111;
hex3=7'b0100001;
hex4=7'b1111001;
hex5=7'b0111111; //display -IdLE-
end
else if(ir1==1'b0)
begin
led1<=1'b1;
led2<=1'b0;
buz<=1'b1;
mot<=1'b1;
hex0=7'b1111111;
hex1=7'b1111111;
hex2=7'b0001001;
hex3=7'b1000001;
hex4=7'b0001001;
hex5=7'b1111111; //display -NULL-
end
else if(ir2==1'b0)
begin
led2<=1'b1;
led1<=1'b0;
buz<=1'b0;
mot<=1'b0;
hex0=7'b1000000;
hex1=7'b1000000;
hex2=7'b1000000;
hex3=7'b0011000;
hex4=7'b0000000;
hex5=7'b0000000; //display 88 9000
end
else if(ir3==1'b0)
begin
led2<=1'b1;
led1<=1'b0;
buz<=1'b0;
mot<=1'b1;
hex0=7'b1000000;
hex1=7'b1000000;
hex2=7'b0110000;
hex3=7'b1111000;
hex4=7'b0011000;
hex5=7'b0000010; //display 69 7300
end
else if(ir4==1'b0)
begin
led2<=1'b1;

```

