BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE 304(January 2023)

Digital Electronics Laboratory

Final Project Report

Section: B1 Group: 03

FPGA-Based Home Automation System

Course Instructors:

Nafis Sadik, Lecturer
Mrinmoy Kundu , Part-Time Lecturer

Signature of Instructor: _____

Academic Honesty Statement:

IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. Type the student ID and name, and put your signature. *You will not receive credit for this project experiment unless this statement is signed in the presence of your lab instructor.*

"In signing this statement, We hereby certify that the work on this project is our own and that we have not copied the work of any other students (past or present), and cited all relevant sources while completing this project. We understand that if we fail to honor this agreement, We will each receive a score of ZERO for this project and be subject to failure of this course."

Signature: _____ **Full Name: Shakir Ahmed** **Student ID: 1906074**	**Signature:** _____ **Full Name: Mehedi Hasan** **Student ID: 1906075**
Signature: _____ **Full Name: Md. Abu Sayed Chowdhury** **Student ID: 1906076**	**Signature:** _____ **Full Name: Md. Sharif Uddin** **Student ID: 1906077**

Table of Contents

1 Abstract

In today's fast-paced world, the concept of smart homes has gained tremendous momentum, providing homeowners with comfort, convenience, and energy efficiency. Home automation improves the lifestyle of the control of the home devices. As the devices are filling the home, the home appliances are filling the homes to improve the comfort to the user. Our project “FPGA-Based Home Automation System," represents a cutting-edge solution designed to transform traditional homes into intelligent, adaptive living spaces. The primary objective of our home automation system is to offer a hands-free, efficient, and user-friendly approach to controlling lighting and ventilation. We have incorporated a range of sensors, including Infrared (IR) sensors and temperature sensors, along with FPGA technology. Here we are using FPGA as controller to control the devices connected to it. We are controlling home appliances using sensors. Instead of employing a micro controller, we are using an FPGA because it can function as both a controller and a processor, allowing us to link many monitorable devices. The choice of FPGA technology over microcontrollers stems from its unparalleled versatility and scalability. This flexibility empowers us to connect and monitor an extensive range of devices, all while utilizing the FPGA as both a controller and a processor.

2 Introduction

Our FPGA-Based Home Automation System is a testament to the fusion of advanced technology with everyday living. It offers us an intelligent and efficient way to manage our lighting and climate control needs, promoting energy savings and enhancing the overall living experience.

Key Features of our FPGA-Based Home Automation System:

Energy Efficiency: To minimize energy wastage, our system automatically turns off lights when no human presence is detected for a predefined period(10s), enhancing both convenience and sustainability.

Fan Control: In response to environmental conditions, specifically when the temperature rises above 30°C, our system activates the fan to maintain a comfortable indoor climate.

Occupancy Tracking: We have implemented a counter variable to keep track of the number of occupants within a room. This variable ensures that even if multiple individuals enter or exit the room, the system maintains the appropriate state without unnecessary switching.

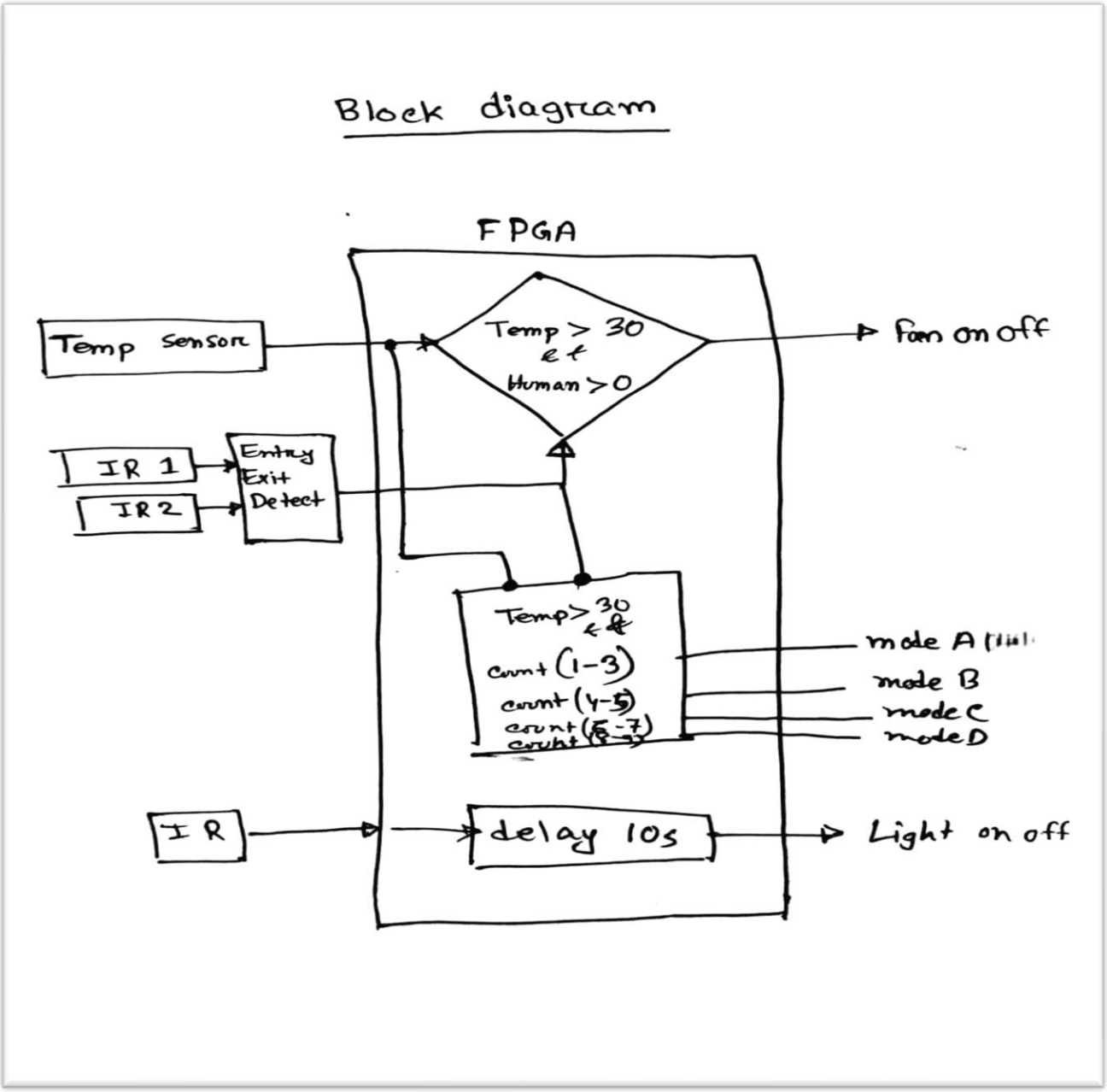
Zero-Human State: When all occupants leave a room, the system resets itself, turning off both lights and fans, while resetting the occupancy counter to zero.

As we go through the project, we will provide a detailed breakdown of its components, functionalities, and the FPGA technology that drives its operation.

3 Design

3.1 Problem Formulation

3.1.1 Identification of Scope


Block diagram
FPGA
Temp sensor
Temp > 30
&
Human > 0
Fan on off
IR 1
IR 2
Entry
Exit
Detect
Temp > 30
&
count (1-3)
count (4-5)
count (6-7)
mode A
mode B
mode C
mode D
IR
delay 10s
Light on off

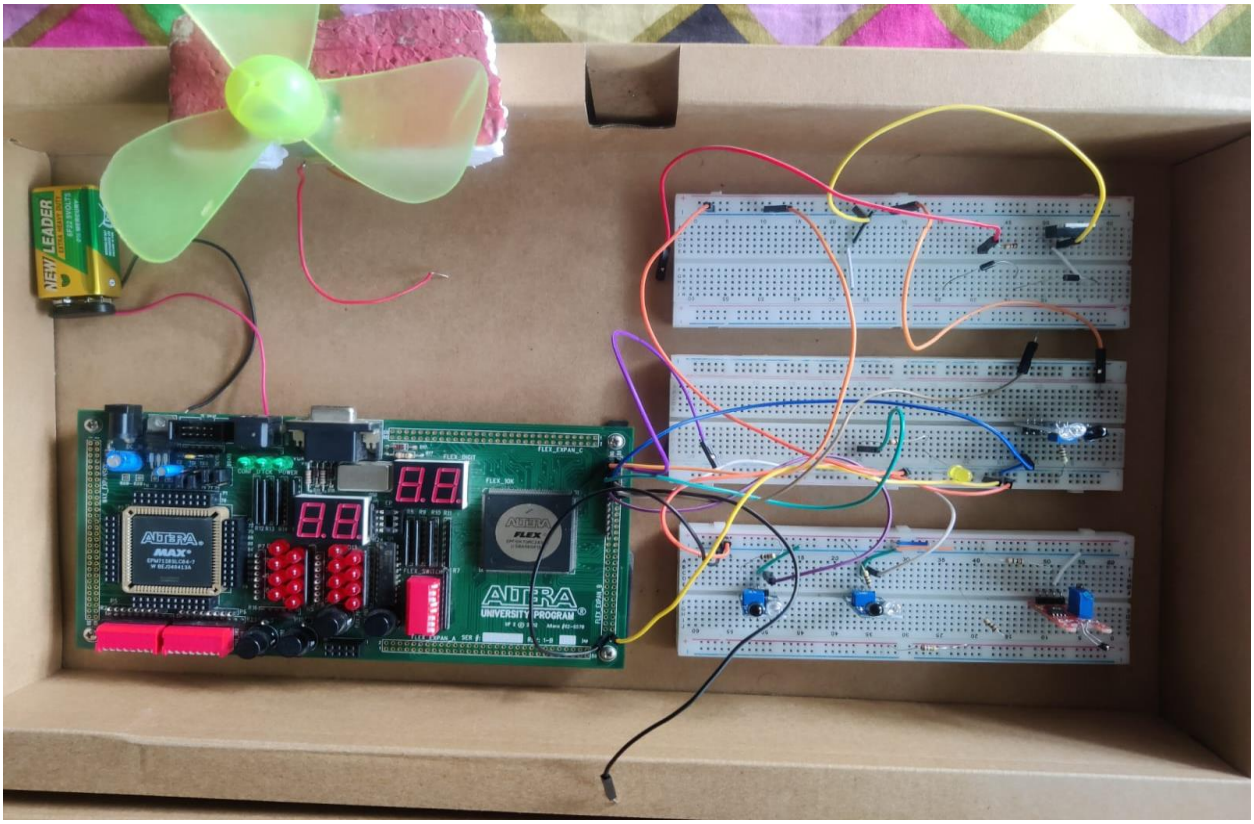
3.1.2 Literature Review

- No human present: Light off
- Human present: IR detected - Light turns on
- Human leaves: light turns off after 10 sec

- No human: Light, fan always off Counter variable=0
- Human enters: Light on Temperature >30°C: Fan on
- More humans enter: Counter variable ++
- Few humans leave: Counter Variable – -
  -if count (1-3) fan speed -mode A
  -if count (4-5) fan speed -mode B
  -if count (6-7) fan speed -mode C
  -if count (8-9) fan speed -mode D

- All human leaves: Counter variable = 0 Light, fan turns off

3.2 Circuit Diagram



3.3 Full Source Code of Firmware

```
module count(input ir, DEV_CLK2, reset1,
		output  led,
		output [6:0] disp2,
		input ir1, ir2, ntc,
		input [2:0] command,
		input reset2,
		output fan,
		output [6:0] people
		);
		wire clk;
		wire clk1;

clk_divisor #(2)inst1(DEV_CLK2,clk);
clk_divisor #(3)inst2(DEV_CLK2,clk1);
lamp1 external_light(ir,clk,reset1,led,disp2);
fan internal_fan(ir1,ir2,ntc,clk1,reset2,fan,people);
endmodule

module lamp1(
		input ir, clk, reset,
		output reg led,
		output reg [6:0] disp2
		);
			reg [31:0] count;
			reg [1:0] state, next;
			localparam idle=2'b00, trig=2'b01, timer=2'b10;

			localparam limit= 32'd62937500, one=32'd6293750, two=32'd12587500, three=32'd18881250;
			localparam four=32'd25175000, five=32'd31468750;
			localparam  six=32'd37762500, seven=32'd44056250, eight=32'd50350000, nine=32'd56643750;

//			localparam limit= 32'd10, one=32'd2, two=32'd4, three=32'd6;
//			localparam four=32'd8, five=32'd10;
//			localparam  six=32'd12, seven=32'd14, eight=32'd16, nine=32'd18;

			always @(posedge clk) begin
				if(reset) state<=idle;
				else state<=next;
				case(state)
					idle: begin
			//disp1<=7'b1111111;

		disp2<=7'b1111111;
							led<=0;
							count<=0;
							if(ir) next <= idle;
							else next<=trig;
							end
					trig: begin

		disp2<=7'b0000001;
							led<=1;
							count<=0;
							if(ir) next<= timer;
							else next<=trig;
							end
		else  begin
timer: begin
							led<=1;
		count<=count+1;
		if(count>limit) begin
		next<=idle;
		disp2<=7'b1110001;
		end
							else if(~ir) next<=trig;
							else begin
		next<=timer;
		if(count<one) begin
			disp2<=7'b0000001;
			end
		else if(count<two) begin
			disp2<=7'b1001111;
			end
		else if(count<three) begin
			disp2<=7'b0010010;
			end
		else if(count<four) begin
			disp2<=7'b0000110;
			end
		else if(count<five) begin
			disp2<=7'b1001100;
			end
		else if(count<six) begin
			disp2<=7'b0100100;
			end
		else if(count<seven) begin
			disp2<=7'b0100000;
			end
		else if(count<eight) begin
			disp2<=7'b0001111;
			end
		else if(count<nine) begin
			disp2<=7'b0000000;
```

```
                        disp2<=7'b0000100;

                else  begin

                        disp2<=7'b0000100;

                        end

                end
                                                        end
                                                endcase
                                        end
endmodule


module fan(
                        input ir1, ir2, ntc,
                        input clk,reset,
                        input [2:0] command,
                        output reg fan,
                        output reg [6:0] person
                        );

                        wire speed25,speed50,speed75,speed100;
                        assign speed100=1;
                        reg [2:0] mode0,mode1,mode2,mode3,mode4;

                        pwm25 m1(clk,speed25);
                        clk_new m2(clk,speed50);
                        pwm75 m3(clk,speed75);

                        reg [1:0] Entrance,ExitOut;
                        reg [3:0] count_in,count_out;
                        reg[3:0] count_net;
reg[31:0] dummy1,dummy2;
                        //reg [1:0] state;
                        localparam Idle=2'd0, Entry=2'd1, Interim1=2'd2,
Room=2'd3;
                        localparam Idle1=2'd0, Exit=2'd1, Interim2=2'd2,
Out=2'd3;

      localparam duration=32'd75500;//delay

                        always @(posedge clk) begin
                                if(reset) begin
                                        Entrance<=Idle;
                                        count_in<=0;
                                        dummy1<=32'd0;
                                        end
                                else begin
                                count_in<=count_in;
        dummy1<=dummy1+32'd1;

                                case(Entrance)
                                        Idle: begin

                if(~ir1&&ir2) Entrance<=Entry;

                else Entrance<=Idle;
                                                        end
                                        Entry: begin

                if(~ir1&&ir2) Entrance<=Entry;



                else if(ir1&&~ir2) Entrance<=Entry;//stability consideration

                else if(~ir1&&~ir2)Entrance<=Interim1;

                else if(ir1&&ir2) Entrance<= Idle;

                else Entrance<=Entrance;


                end
                        end
                                                                        else
if(~ir1&&~ir2) Entrance<=Interim1;

                else if(ir1&&ir2) Entrance<=Idle;

                else if(ir1&&~ir2) begin

                 Entrance<= Room;

                 if (dummy1>=duration) begin

                                        count_in<=count_in+3'd1;

                                        dummy1<=0;

                                        end

                                else count_in<=count_in;end

                else Entrance<=Entrance;

                end
                                                Room: begin

                if(ir1&&ir2) Entrance<=Idle;

                else if(~ir1&&ir2) Entrance<=Entry;

                else if(~ir1&&~ir2) Entrance<=Interim1;

                else if(ir1&&~ir2) Entrance<=Room;

                else Entrance<=Entrance;

                end
                                                endcase
                                        end
                                        end
                                always @(posedge clk) begin
                                        if(reset) begin
                                                ExitOut<=Idle1;
                                                count_out<=0;
                                                dummy2<=32'd0;
                                                end
                                        else begin
                                        count_out<=count_out;
        dummy2<=dummy2+32'd1;

                                        case(ExitOut)
                                                Idle1: begin

                if(ir1&&~ir2) ExitOut<=Exit;

                else ExitOut<=Idle1;
                                                                        end
                                                Exit: begin

                if(~ir1&&ir2) ExitOut<=Exit;



                else if(ir1&&~ir2) ExitOut<=Exit;//stability consideration

                else if(~ir1&&~ir2) ExitOut<=Interim2;

                else if(ir1&&ir2) ExitOut<= Idle1;

                else ExitOut<=ExitOut;
```

```
                                        Interim1: begin
            if(~ir1&&ir2) Entrance<=Entry;
Interim2: begin
            if(ir1&&~ir2) ExitOut<=Exit;
            else if(~ir1&&~ir2) ExitOut<=Interim2;
            else if(ir1&&ir2) ExitOut<=Idle1;
            else if(~ir1&&ir2) begin
             ExitOut<= Out;
             if (dummy2>=duration) begin
                                    count_out<=count_out+3'd1;
                                    dummy2<=0;
                                    end
              else count_out<=count_out;end
            else ExitOut<=ExitOut;
            end
                        Out: begin
            if(ir1&&ir2) ExitOut<=Idle1;
            else if(ir1&&~ir2) ExitOut<=Exit;
            else if(~ir1&&~ir2) ExitOut<=Interim1;
            else if(~ir1&&ir2) ExitOut<=Out;
            else ExitOut<=ExitOut;
            end
                                                endcase
                                    end
                                    end
                        always @(*) count_net=count_in-count_out;
                        always @(*) begin
                        mode0=0;
                        case(command)
                                    3'b001: begin
            mode1=speed25;
            mode2=speed25;
            mode3=speed25;
            mode4=speed25;
            end
                                    3'b010: begin
            mode1=speed50;
            mode2=speed50;
            mode3=speed50;
            mode4=speed50;
            end
                        3'b100: begin
            mode1=speed75;
            mode2=speed75;
mode3=speed75;
            mode3=speed75;  end
                                                end
3'b011: begin
            mode1=speed100;
            mode2=speed100;
            mode3=speed100;
            mode3=speed100;
            end
                                    3'b101: begin
            mode1=speed25;
            mode2=speed50;
            mode3=speed75;
            mode3=speed100;
            end
                                    3'b110: begin
            mode1=speed50;
            mode2=speed50;
            mode3=speed100;
            mode3=speed100;
            end
                                    default: begin
            mode1=speed50;
            mode2=speed50;
            mode3=speed50;
            mode3=speed50;
            end
                                    endcase
                                                if(ntc) begin
            if(count_net==0) fan=mode0;
            if(count_net>0 && count_net<=3)
            fan=mode1;                                          else if
(count_net>3 && count_net<=6)
            fan=mode2;                                          else if
(count_net>6 && count_net<8)
            fan=mode3;                                          else
fan=mode4;
                                                                end
                                                    else fan=0;
                                    end
                                    always @(*) begin
                                                case(count_net)
```

```
        4'd0:person=7'b0000001;

        4'd1:person=7'b1001111;

        4'd2:person=7'b0010010;

                                                end
                                end
4'd3:person=7'b0000110;

        4'd4:person=7'b1001100;

        4'd5:person=7'b0100100;

        4'd6:person=7'b0100000;

        4'd7:person=7'b0001111;

        4'd8:person=7'b0000000;

        4'd9:person=7'b0000100;

        default:person=7'b1111111;
                                endcase
                                end
endmodule


module pwm25(input clk, output reg pwms);
                        reg [3:0] count=1'd0;
                        reg [31:0] count1=32'd0;
                        wire clk_f;
                        clk_new ins(clk,clk_f);
                        parameter A=1'b0,B=1'b1;
                        always@(posedge clk_f) begin
                                count1<=count1+1;
                                if(count1<32'd10000)
                                pwms<=B;
                                else begin
                                                count1<=count1;
                                                count<=count+4'd1;
                                                case(count)

        5,6: pwms<=B;

        7: begin

        pwms<=B;

        count<=0;

        end

        default:pwms<=A;
                                                                endcase
                                                end
                                end
                        endmodule
module pwm75(input clk, output reg pwms);
                        reg [3:0] count=1'd0;
                        reg [31:0] count1=32'd0;
                        wire clk_f;
                        clk_new ins(clk,clk_f);
                        parameter A=1'b1,B=1'b0;
                        always@(posedge clk_f) begin
                                count1<=count1+1;
                                if(count1<32'd50)
                                pwms<=B;
                                else begin
                                                count1<=count1;
                                                count<=count+4'd1;
                                                case(count)
                                6: pwms<=B;
                                7: begin

        pwms<=B;

        count<=0;

        end

        default:pwms<=A;
                                                                endcase
endmodule

module clk_new(input clk,output clk_final);
        wire one,two,three,four,five;
        clk_divisor first(clk,one);
        clk_divisor second(one,two);
        clk_divisor third(two,three);
        clk_divisor fourth(three,four);
        clk_divisor fifth(four, five);
        clk_divisor fina(five,clk_final);
endmodule



module clk_divisor #(parameter divisor=16'd2)(
        input clk_in,
        output reg clk_out
        );
                reg[15:0] count=16'd0;
                always@(posedge clk_in) begin
                        count<=count+16'd1;
                        if(count>=divisor-16'd1) begin
                                count<=0;
                                clk_out<=~clk_out;
                                end
                        else clk_out<=clk_out;
                        end
                endmodule
```

Table: Source Code for the main program

The source code of our Home Automation System, implemented in Verilog, serves as the digital blueprint that controls the behavior of the FPGA (Field-Programmable Gate Array). It defines the logic for managing various home devices, sensors, and user interactions. The code orchestrates tasks such as detecting human presence, monitoring environmental conditions, and controlling lights and fans based on predefined conditions. It's structured as a Finite State Machine (FSM) to model complex behaviors efficiently. The Verilog source code is the heart of our system, enabling it to respond intelligently to user needs and environmental factors, ensuring seamless automation and enhancing the overall living experience.

4 Implementation

4.1 Description

In this section, we provide an in-depth description of the implementation phase of our Home Automation project. We outline the key components, technologies, and methodologies used to bring our system to life. This includes details about the hardware, software, and network infrastructure deployed.

4.2 Experiment and Data Collection

This subsection delves into the experimentation and data collection processes. We describe how the Home Automation System was deployed in a real-world environment, including any challenges encountered during installation. Additionally, we discuss the methods employed to collect data regarding device usage, energy consumption, and user interactions.

4.3 Data Analysis

Data analysis is a crucial aspect of our project's implementation. In this section, we elaborate on the techniques and tools used to process, analyze, and interpret the collected data. We also discuss the insights gained from the analysis, such as user behavior patterns, energy usage trends, and system performance.

4.4 Results

The "Results" subsection presents the outcomes of our Home Automation System's implementation. We highlight the system's effectiveness in achieving its intended goals, such as energy savings, convenience, and user satisfaction. Any notable findings, improvements made during the implementation phase, and their impacts on the project are discussed in detail..

5 Design Analysis and Evaluation

5 .1 Novelty

In our project, we want to control the entry and exit of a person .When a man is in the room ,the light and fan both turn on automatically if the sufficient conditions are met . On the contrary, when the man is out of the room ,the light and fan turn off . Hence the energy wastage is reduced automatically. Besides in our project, we control the timing of lamp-post . After passing by any object ,the lamp-post will turn on and turn off after 10s.

5 .2 Design Considerations

Outside room

- No human present: Light off
- Human present: Light turns on
- Human leaves: light turns off after 10 sec

Inside room

- No human: Light, fan always off
- Counter variable=0
- Human enters: Light on
- Temperature >30°C: Fan on
- More humans enter: State does not change.
- Counter variable ++
- Few humans leave: State does not change
- Counter Variable --
- All human leaves:
- Counter variable = 0
- Light, fan turns off

5.3Investigations

5.3.1 Literature Review

In this section, we will conduct a comprehensive review of existing literature related to home automation systems, FPGA technology, and the ethical, legal, and societal implications of such systems. We will explore the current state of home automation, including advancements,

challenges, and trends. Additionally, we will examine relevant research on FPGA-based solutions and their applications in the field. The literature review will provide a foundation for understanding the context and existing knowledge related to our project.

5.3.2 Experiment Design

This section will outline the experimental methodology for implementing and testing our "FPGA Based Home Automation System." We will detail the hardware and software components used, including the FPGA board, sensors, actuators, and programming languages (e.g., Verilog or VHDL). We will describe the setup for motion detection, temperature regulation, and fan control based on user presence and climate conditions. We will also explain the ethical considerations, data privacy measures, and safety precautions integrated into the design. The experiment design section will provide a clear roadmap for the practical implementation of our project.

5.3.3 Data Analysis and Interpretation

Upon successful implementation, this section will focus on the analysis and interpretation of the data collected during the experiment. We will present quantitative and qualitative findings related to system performance, energy efficiency, user experience, and ethical considerations. We will use appropriate data visualization techniques and statistical methods to analyze the results. Ethical, legal, and societal implications will be discussed in light of the findings. The data analysis and interpretation section will conclude with insights, conclusions, and recommendations based on the experiment's outcomes.

5.4 Limitations of Tools

- Accurate human count
- IR sensor accuracy
- Timing accuracy
- NTC thermistor value was not stable enough

5.5 Impact Assessment

5.5.1 Assessment of Societal and Cultural Issues

It is essential to assess the potential societal and cultural issues that may arise. Some considerations are given below.

Privacy Concerns: Home automation systems have access to personal data about occupants' routines and behaviors. Ensure data privacy by implementing robust security measures to protect against unauthorized access and data breaches.

Accessibility and Inclusivity: Ensure that the system is accessible and usable by individuals with disabilities. Consider how voice control or other features may benefit users with mobility or sensory impairments.

Cultural Sensitivity: Be aware of cultural differences and sensitivities while designing user interfaces and voice command systems. Avoid potential issues related to culturally insensitive language or imagery.

Technological Accessibility: Ensure that the system is not overly complex or reliant on expensive technology, making it accessible to a broad range of users across different socioeconomic backgrounds.

Ethical AI and Automation: Ethical considerations may arise when using artificial intelligence algorithms for decision-making. Ensure fairness and transparency in the system's logic to avoid bias and discrimination.

Sustainability:
Consider the environmental impact of the system's components and encourage the use of eco-friendly materials and practices.

5.5.2 Assessment of Health and Safety Issues

It is necessary to ensure that the system operate without posing any risks to users or the environment. Here are some health and safety considerations to keep in mind:

Electrical Safety: Ensure that all electrical connections, including those to the FPGA board, sensors, and actuators, are correctly wired and insulated to prevent electrical hazards, such as shocks or short circuits.

Fire Safety: When working with electrical components, be aware of fire hazards. Use appropriate wiring, fuses, and circuit breakers to prevent overheating and electrical fires.

Device Safety: Ensure that all devices connected to the system, such as servos, LEDs, and sensors, are in good working condition and meet safety standards to prevent malfunctions or accidents.

Temperature Control: System controls heating or cooling devices maintain safe temperature levels to prevent overheating or discomfort for occupants.

Sensor Accuracy: Regularly calibrate and maintain sensors like the temperature sensor to ensure accurate readings. Incorrect temperature readings could lead to unsafe conditions, especially if the fan control is based on temperature.

5.5.3 Assessment of Legal Issues

- Ensure the product safety, complies with warranty and guarantee requirements, and meets labeling and product safety standards.
- Comply with any environmental regulations related to the disposal and recycling of electronic components used in your system.

5.6 Ethical Issues

It is imperative to uphold ethical considerations to align our project with moral principles and societal values. Key ethical concerns encompass privacy and data security, necessitating stringent measures to safeguard user information from unauthorized access or breaches. Obtaining informed consent from users, transparently explaining data collection and usage, is paramount. Additionally, ensuring accessibility for individuals with disabilities and promoting fairness by mitigating algorithmic bias is crucial. Moreover, the project should strive to minimize its environmental impact through sustainable practices and energy-efficient components and operations, thus embodying a holistic commitment to ethical integrity.

6 Reflection on Individual and Team work

a. Individual Contribution of Each Member

Shakir(1906074)

-Verilog code implementation.
-Debug the code.
-Purchse equipment.

Mehedi(1906075)

-Hardware setup.
-Making FSM.

-Purchase equipment.

Sayed(1906076)

-Verilog code implementation.
-Debug the code
-Idea co-ordination.

Sharif(1906077)
-Making FSM.
-Hardware setup.
-Management.

6.2 Mode of TeamWork

1.Hardware implementation
2.Assembling
3.Debugging
4.Setup building
5.Report and Presentation

b. Log Book of Project Implementation

Date	Milestone achieved	Team Role	Comments
19.06.23	Project Proposal Discussion & Assignment	Whole team discussion in lab	Proposed Home Automation
03.07.23	Project Proposal Presentation	Presented a brief idea and working procedure of the project	Speed control of fan introduced
10.07.23	Online meeting & budgeting	Whole team attend a zoom to decide the next working procedure	A tentative cost sheet prepared
17.07.23	Buy and set up Process	Two members of the team bought the components and others started to set up the components.	Primary set up completed
24.07.23	First Progress Presentation	Team presented the first presentation	Lamp post light work smoothly
31.07.23	Fan code	Team members discussed	Failed
07.08.23	Fan Code	Some set the circuit others worked with code	Fan code did not work
14.08.23	Fan	same	Work properly
21.08.23	Cascading	same	Cascading failed
04.09.23	Cascading	All team members worked together	Debugging succeed
11.09.23	Final Presentation		

7 Communication

Effective communication of our project is crucial to its success. Start by clearly defining our project's goals, objectives, and the problem it aims to solve. Develop a compelling narrative that highlights the project's significance and its potential impact on users' lives. Utilize various communication channels, such as presentations, reports, and user manuals, to convey technical details, benefits, and safety considerations. Engage with your target audience by addressing their needs and concerns, and be prepared to answer questions and provide demonstrations. Consider leveraging visual aids, prototypes, and user-friendly language to simplify complex concepts. Regular updates and feedback sessions can also foster a sense of collaboration and trust with stakeholders. Ultimately, effective communication ensures that your project's value is well understood and appreciated by your intended audience.

7.1 Executive Summary

The "FPGA Based Home Automation System" is an innovative and user-friendly solution that leverages FPGA technology to enhance home automation. Our project focuses on energy efficiency, privacy, and user control. It includes motion detection for lighting, temperature regulation, and fan control based on user presence and climate conditions. Our system prioritizes data privacy and security, offering a seamless and accessible user experience. We emphasize environmental responsibility, ensuring that the system minimizes energy consumption. This project not only introduces novelty but also addresses societal, ethical, and legal considerations, providing users with a safe, efficient, and customizable home automation experience.

7.2 User Manual

The User Manual for our FPGA-Based Home Automation System is designed to provide clear and concise guidance to users, ensuring a safe and efficient experience. It commences with an introduction, giving an overview of the system and its purpose. Safety precautions are emphasized, covering electrical and fire safety, child safety, and emergency procedures. The manual guides users through system setup, hardware installation, and initial software configuration. Detailed instructions on using the system's interface, controlling lights and appliances, and setting timers are included. Users can explore advanced features, customization options, and energy efficiency tips. Troubleshooting guidance and common FAQs are provided for problem-solving. Maintenance and care recommendations, data privacy and security information, and user support contacts are also outlined. The manual concludes with details on regulatory compliance, a glossary, and appendices for reference materials. This user manual aims to ensure that users have a seamless and secure experience with our home automation system.

8 Project Management and Cost Analysis

8.1 Bill of Materials

Description	Quantity	Cost
FPGA	1	000 (From lab)
USB blaster	1	780
IR Sensor	4	350
DC motor	1	70
Battery (9V)	1	60
NTC Thermistor Module	2	180
Resistor	1set	20
Diode	5	10
BJT(BD135)	2	20
LED	10	20
Jumper Wire	1 set	40
Bread Board	3	180
Total		1730

9 Future Work

Our project can be modified in future in various ways.

Voice Recognition: Implement voice control to provide hands-free operation of lights and fans. Integrate a speech recognition module to interpret voice commands.
Mobile App Integration: Develop a mobile application that allows users to control and monitor the home automation system remotely using smartphones or tablets
Security: Enhance security by integrating cameras and sensors for intrusion detection. Implement notification systems to alert homeowners of security breaches.
User Profiles: Create user profiles that allow different users to have customized settings and permissions within the home automation system.
Voice Assistance: Integrate virtual voice assistants like Amazon Alexa or Google Assistant for natural language interaction with the system.
Expandable Sensor Network: Add more sensors for additional functionality, such as gas leakage detection, smoke detection, or smart door locks.
Remote Monitoring and Control: Enable remote monitoring and control of household appliances, such as ovens, refrigerators, or coffee makers, for added convenience.
Integration with Smart Devices: Ensure compatibility with existing smart home devices and platforms such as Apple HomeKit, Google Home, or Amazon Alexa.

10 References

1. **https://ieeexplore.ieee.org/document/7453813**

2. **https://www.ijert.org/home-automation-through-fpga-controller**

3. **https://www.researchgate.net/publication/313386108_IOT_based_home_automation_using_FPGA**

4. **https://www.google.nl/books/edition/FPGA_Based_Embedded_System_Developer_s_G/xGpQDwAAQBAJ?hl=en&gbpv=1&dq=fpga+based+home+automation+system&pg=PA217&printsec=frontcover**