HOME AUTOMATION SYSTEM

A Mini Project Report

submitted by:

Abhishek Subramanian

Dhineshkumar V

Suriya Ramesh

Yogaram Shos

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

DEPARTMENT OF
ELECTRONICS AND COMMUNICATION ENGINEERING
ANNA UNIVERSITY, CHENNAI

REPORT ON

HOME AUTOMATION SYSTEM

Abhishek Subramanian (2018105002)

Dhineshkumar V (2018105522)

Suriya Ramesh (2018105064)

Yogaram Shos (2018105622)

INDEX

INTRODUCTION

The main objective of this project is to develop a home automation system that detects motion and turns the devices in a room on/off based on the presence of people in the room. In the modern era as technology is advancing, our houses are also getting smarter. Modern houses are gradually shifting from conventional switches to various other alternatives. The primary reason behind the replacement of conventional switches is that, conventional switches are located in different parts of the house which make it cumbersome for the average person to operate each time he/she wants to turn a device on/off.

Another reason why such alternatives make sense is when you look at the wastage of energy caused because a person forgets to turn the light, fan, air-conditioning system, etc. off when he/she leaves the room. In order to overcome the above limitations of conventional switches, we can use motion detected sensors that sense when a person enters/ leaves a room and turns the various devices on/ off respectively. This massively reduces human effort and interaction while also saving valuable energy.

This is implemented with the help of two Passive Infrared Sensors (PIR), one outside and one inside the room. When a person enters a room, the sensor outside the room first gets activated followed by the sensor lying inside the room. This results in the devices in the room automatically switching on eliminating the need to manually turn them on. If a user decides to turn particular device off, he/ she can override the automation system and manually do so. Similarly, when a person leaves the room, the sensor inside the room first gets activated and then the sensor outside the room gets activated.

On the small scale, this is implemented with the help of an Artix – 7 based FPGA Development Board (Basys 3) from Xylinx and two HC-SR501

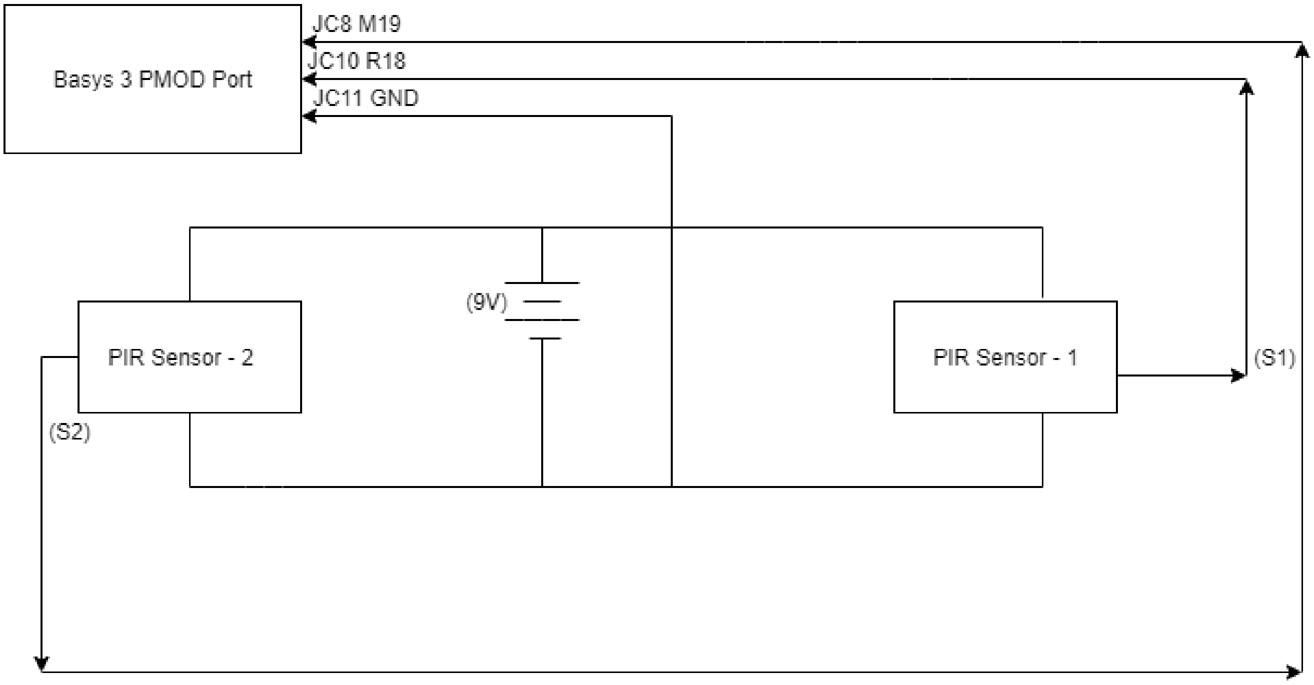
Passive Infrared sensors. These sensors are placed in a manner such that they are not facing each other. When the Passive Infrared sensor placed on the right detects motion, it turns two LEDs on. When the Passive Infrared sensor on the left detects motion, it turns the two LEDs off.

The two Passive Infrared sensors have a voltage requirement of 4.5V – 12V each, in order to operate. Since the Basys 3 board can only output a maximum voltage of 3.3V, the two sensors are powered with the help of an external DC source.

When the Passive Infrared sensor detects motion, it sends out a digital high signal of 3V. When it does not detect any motion the output voltage drops to 0V. A Verilog code is written on Vivado to implement the above logic.

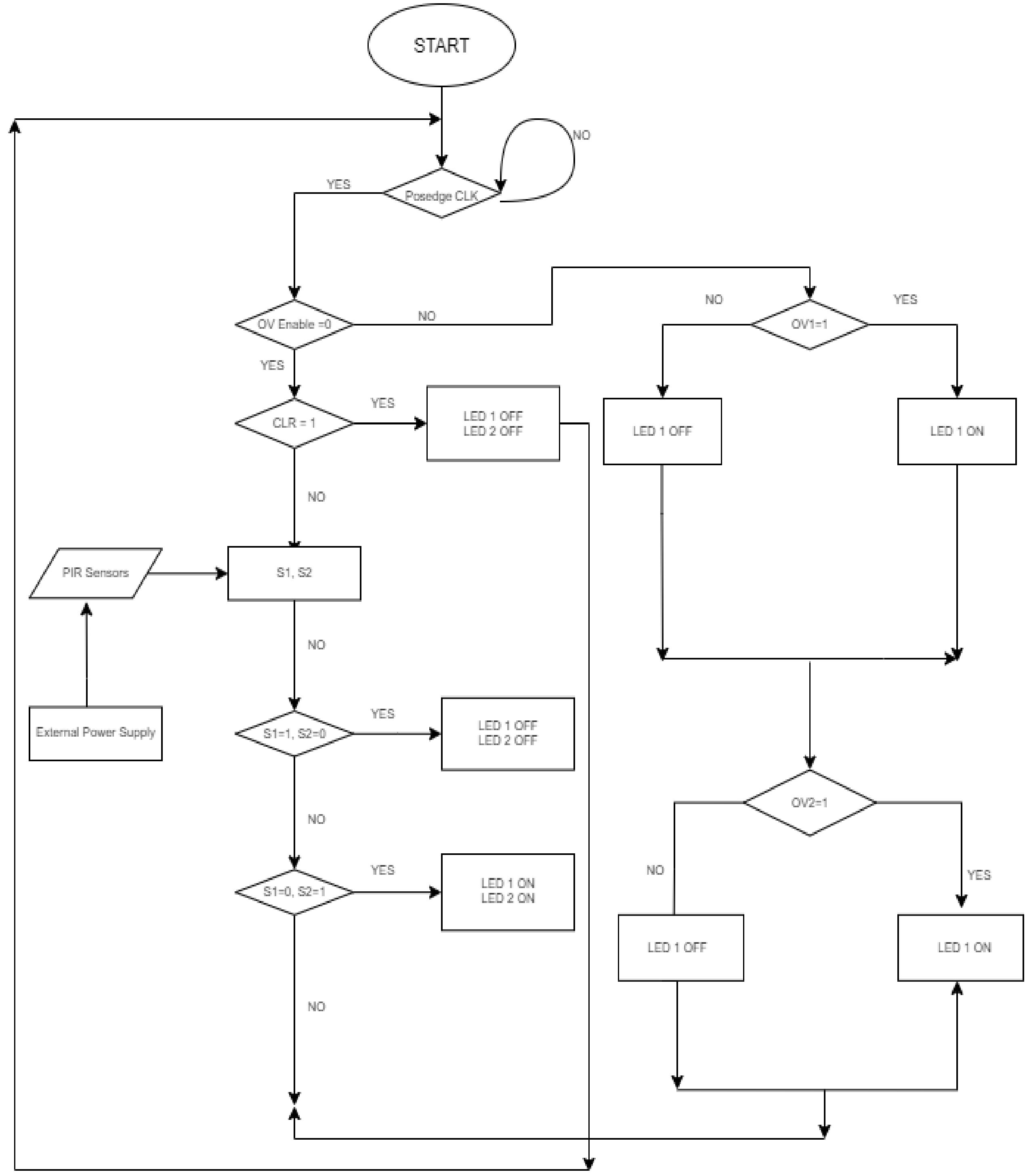
METHOD

Block Diagram

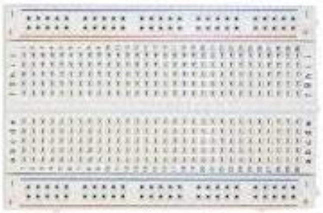


- The PIR sensors are connected to the appropriate PMOD ports on the Artix – 7 Based FPGA development board (Basys 3) according to the above block diagram.

Flowchart


START
Posedge CLK
NO
YES
OV Enable =0
NO
YES
OV1=1
NO
YES
CLR = 1
YES
LED 1 OFF
LED 2 OFF
LED 1 OFF
LED 1 ON
NO
PIR Sensors
S1, S2
NO
External Power Supply
S1=1, S2=0
YES
LED 1 OFF
LED 2 OFF
OV2=1
NO
NO
YES
S1=0, S2=1
YES
LED 1 ON
LED 2 ON
LED 1 OFF
LED 1 ON
NO

Components Used

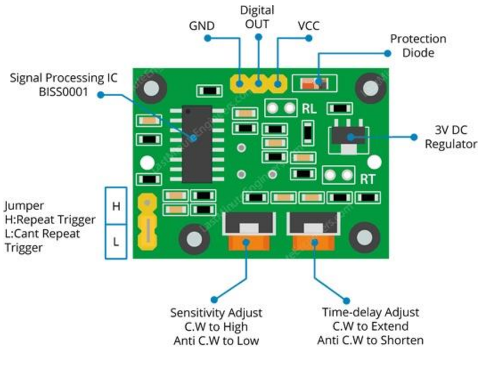
Sr. No.	Name of the Component
1	Artix – 7 FPGA Development Board (Basys 3)
2	2 HC-SR501 Passive Infrared Sensors
3	Breadboard 
4	9 Volt Alkaline Battery
5	Connecting Wires
6	Micro USB Cable

Artix – 7 based FPGA Development Board (Basys 3)

- The Basys 3 is an entry-level FPGA development board designed exclusively for the Vivado Design Suite featuring the Xilinx Artix-7 FPGA architecture.
- It consists of 33,280 logic cells in 5200 slices, 1800 Kbits of fast block RAM, 5 clock management tiles, 90 DSP slices and an Analog to Digital Convertor (ADC). It can also support internal clock speeds exceeding 450 MHz.
- It includes 16 programmable switches and LEDs, 5 programmable switch buttons, a USB port, a 4-digit seven segment display amongst various other peripherals.

HC-SR501 Passive Infrared (PIR) Sensor

- A PIR sensor is specially designed to detect Infrared Radiation. It consists of two main parts: A Pyroelectric Sensor and a special lens called Fresnel lens which focuses the infrared signals onto the pyroelectric sensor.
- The HC-SR501 PIR sensors are low power, low cost sensors that are rugged and easy to interface with. It has three terminals: VCC, Output and Ground as shown in the diagram below. It has a built-in voltage regulator so it can be powered by any DC voltage from 4.5 to 12 volts.


GND
Digital OUT
VCC
Protection Diode
Signal Processing IC BISS0001
RL
3V DC Regulator
RT
Jumper H:Repeat Trigger L:Cant Repeat Trigger
H
L
Sensitivity Adjust C.W to High Anti C.W to Low
Time-delay Adjust C.W to Extend Anti C.W to Shorten

Software Used

Xylinx Vivado 2019.1

- Vivado Design Suite is a software suite produced by Xilinx for synthesis and analysis of HDL designs, superseding Xilinx ISE additional features for system on a chip development and high-level synthesis. Vivado represents a ground-up rewrite and re-thinking of the entire design flow when compared to the dated ISE Design Suite.

- Vivado was introduced in April 2012. It is an Integrated Design Environment with system-to-IC level tools built on a shared scalable data model and a common debug environment. Vivado includes Electronic System Level design tools for synthesizing and verifying C-based algorithmic IP, standards-based packaging of both algorithmic and RTL IP for reuse, standards-based IP stitching and systems integration of all types of system building blocks and the verification of blocks and systems. A free version Web Pack Edition of Vivado provides designers with a limited version of the design environment, which is what was used to develop this project.

- A variety of design sources are supported including RTL Designs and Netlist Designs. The Vivado Design Suite implementation is a timing-driven flow. It supports industry standard Synopsys Design Constraints (SDC) commands to specify design requirements and restrictions, as well as additional commands in the Xilinx Design Constraints format (XDC).

- We can implement the design using Verilog or VHDL. A wide variety of boards can be selected for the program to run on. This project has been implemented on the Artix – 7 based FPGA

Development board (Basys 3) using Verilog as the Hardware Description Language.

Verilog

- Verilog is a Hardware Description Language (HDL). It is a language used for describing a digital system like a network switch or a microprocessor or a memory or a flip−flop. It means, by using an HDL we can describe any digital hardware at any level. Designs, which are described on HDL are independent of technology, very easy for designing and debugging, and are normally more useful than schematics, particularly for large circuits.
- Verilog supports a design at many levels of abstraction. The major three are major levels are:
  - Behavioral Level
  - Register Transfer Level
  - Gate level

Behavioral level

- This level describes a system by concurrent algorithms (Behavioral). Every algorithm is sequential, which means it consists of a set of instructions that are executed one by one. Functions, tasks and blocks are the main elements. There is no regard to the structural realization of the design.

Register Transfer Level

- Designs using the Register−Transfer Level specify the characteristics of a circuit using operations and the transfer of data between the registers. Modern definition of an RTL code is "Any code that is synthesizable is called RTL code".

Gate level

- Within the logical level, the characteristics of a system are described by logical links and their timing properties. All signals are discrete signals. They can only have definite logical values (`0', `1', `X', `Z`). The usable operations are predefined logic primitives (basic gates). Gate level modelling may not be a right idea for logic design. Gate level code is generated using tools like synthesis tools and his netlist is used for gate level simulation and for backend.
- Verilog has built-in primitives like logic gates, transmission gates and switches. These are rarely used for design work but they are used in post synthesis world for modelling of ASIC/FPGA cells.

Gate level modelling exhibits two properties –

Drive strength − The strength of the output gates is defined by drive strength. The output is strongest if there is a direct connection to the source. The strength decreases if the connection is via a conducting transistor and least when connected via a pull-up/down resistive. The drive strength is usually not specified, in which case the strengths defaults to strong 1 and strong 0.

Delays − If delays are not specified, then the gates do not have propagation delays. If two delays are specified, then first one represents the rise delay and the second one, fall delay; if only one delay is specified, then both, rise and fall are equal. Delays can be ignored in synthesis.

RESULT

The setup consists of two Passive Infrared (PIR) sensors. As mentioned, it consists of three pins – VCC, GND and Output. Since the Artix – 7 FPGA Development Board (Basys 3) can only provide a maximum output of 3.3V, we use an external DC power supply, which is of 9V.

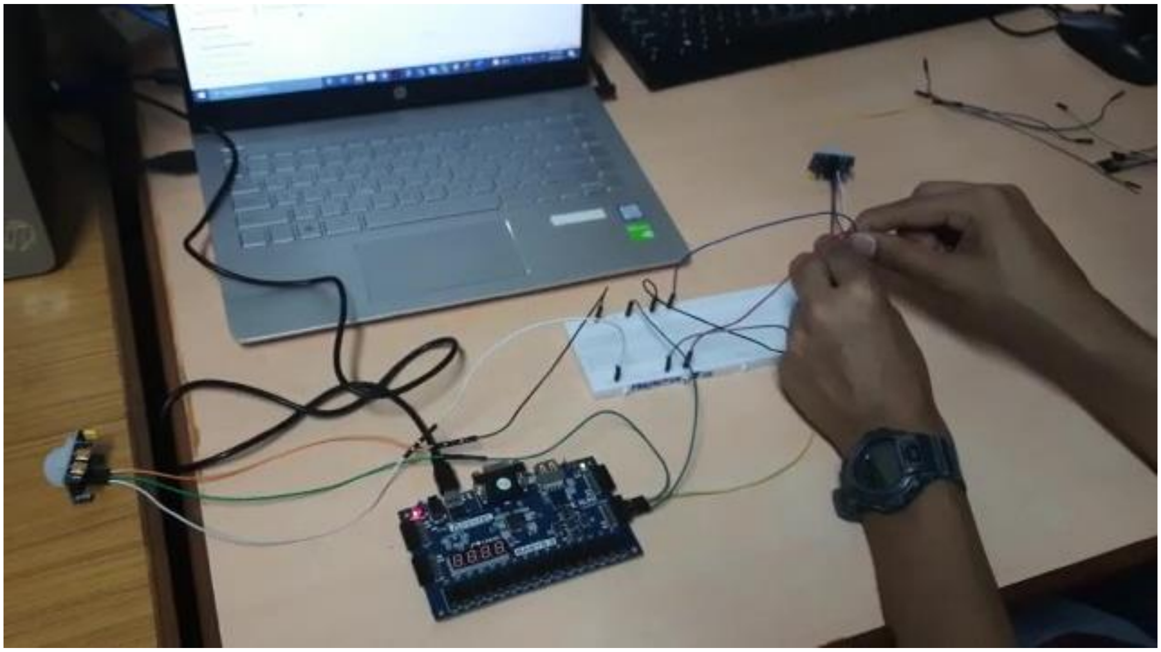
The VCC terminals of the Passive Infrared (PIR) sensors are connected to the positive terminal of the power supply with the help of a breadboard. The GND terminals of the Passive Infrared (PIR) sensors are connected to the negative terminal of the battery which is in turn connected to the GND terminal of the Artix – 7 FPGA Development Board (Basys 3) in order to complete the circuit.

The output terminals from the Passive Infrared (PIR) sensors are connected to the PMOD ports on the board. The output terminal from the first PIR sensor, which turns the LEDs on is connected to port ‘M19’ while the output from the other PIR sensor is connected to port ‘R18’.

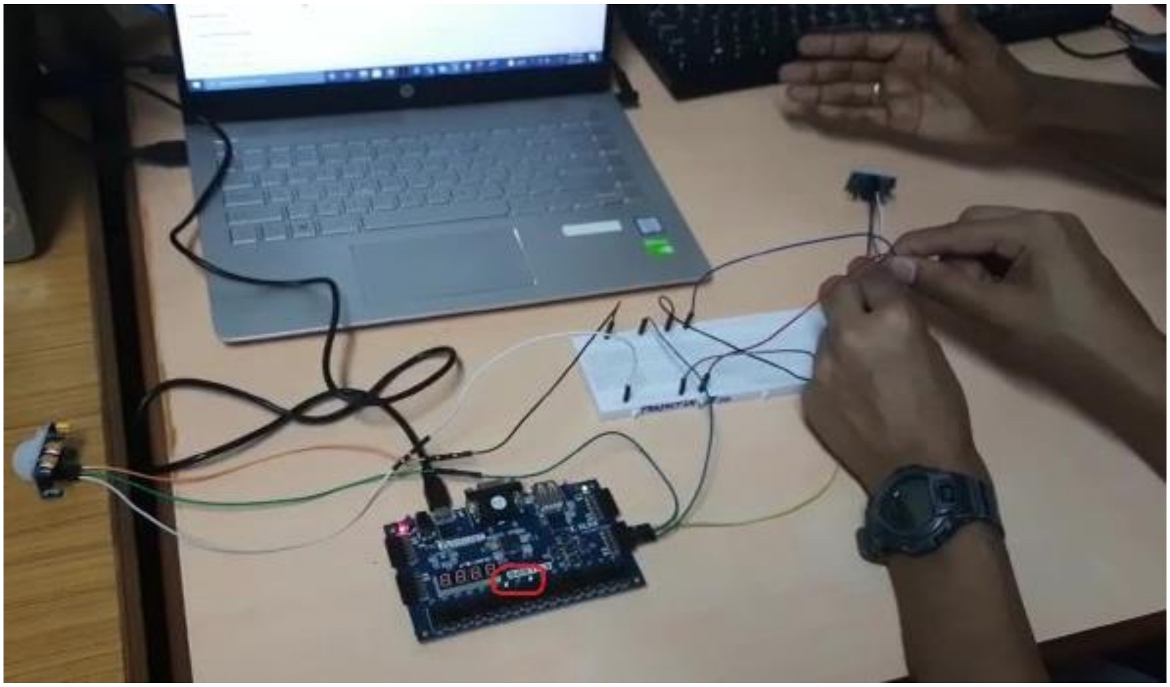
When the PIR sensor connected to port ‘M19’ detects motion, it sends a voltage high signal which turns the two LEDs (V4 and V13) on. The PIR sensor connected to port ‘R18’ upon detecting motion turns the LEDs off.

Setup

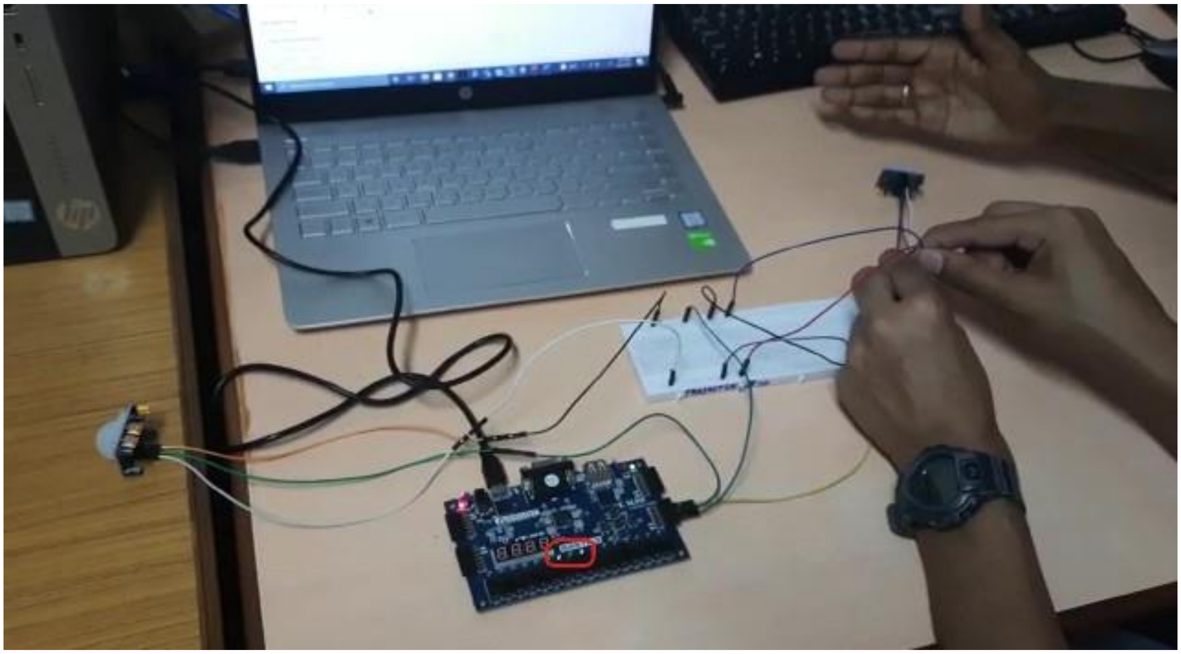
- The PIR sensors do not detect any motion thus resulting in the LEDs not turning on.



- LEDs turn on when the PIR sensor connected to port M19 on the board detect motion.

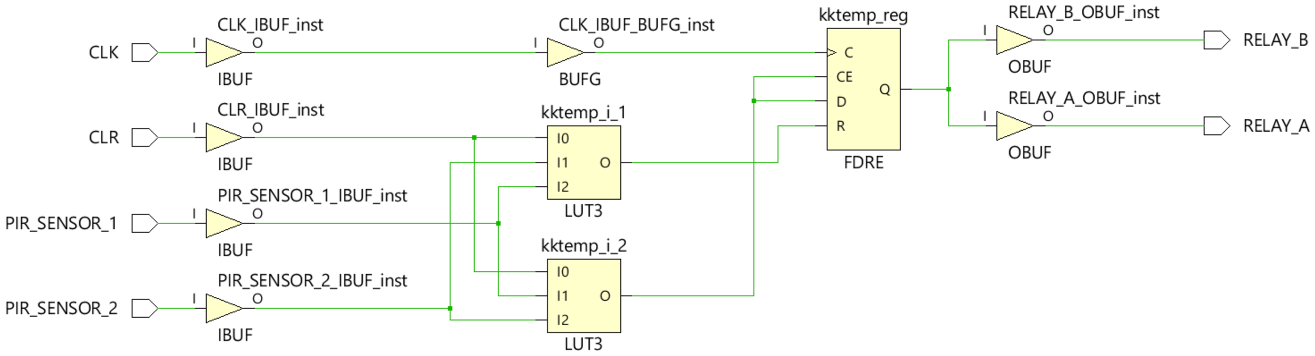


- LEDs turn off when the PIR sensor connected to port R18 detects motion.

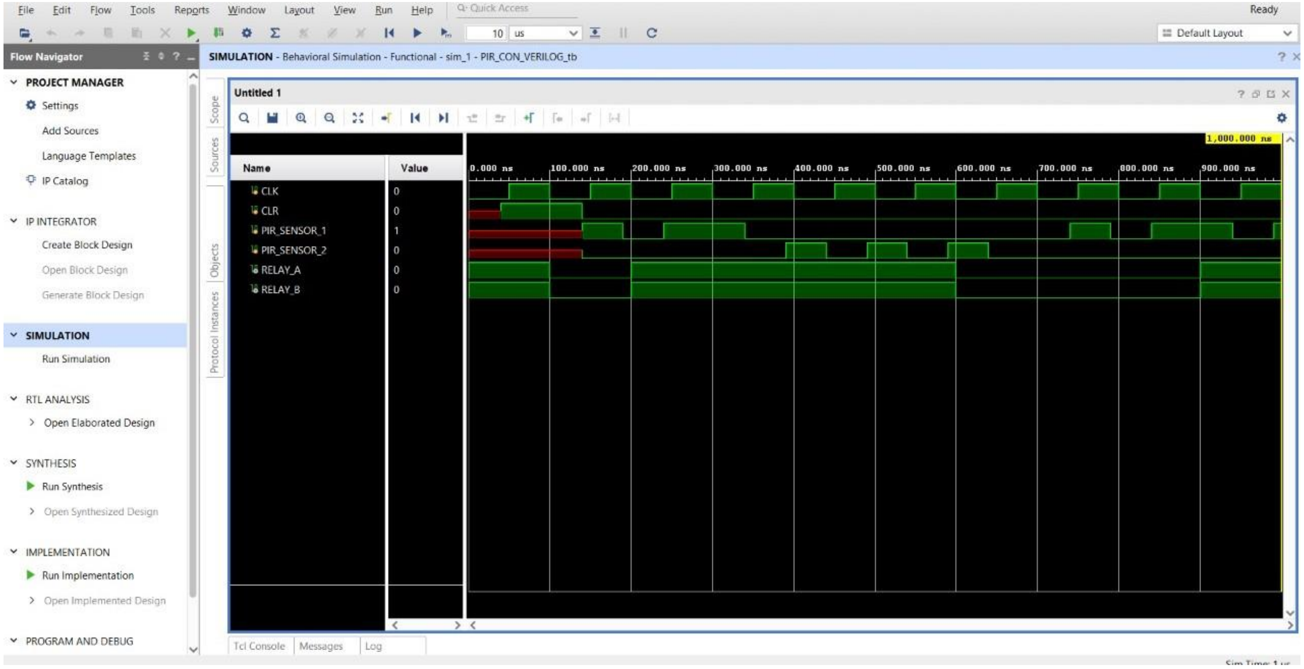


- Once the PIR sensor connected to port M19 detects motion, the LEDs remain on as long as the PIR sensor connected port R18 detects motion, turning them off. This emulates the scenario wherein the devices in a room stay on as long as there are people present in the room. The devices will turn off only when the people leave the room.

Schematic



Simulation



SIMULATION - Behavioral Simulation - Functional - sim_1 - PIR_CON_VERILOG_tb

Name	Value
CLK	0
CLR	0
PIR_SENSOR_1	1
PIR_SENSOR_2	0
RELAY_A	0
RELAY_B	0

CONCLUSION

We perform a number of monotonous tasks in our daily lives which can easily be replaced with the help of some intelligent sensors and a capable FPGA board. In the modern era, we have already automated a number of such tasks with the help of sensors ranging from the sliding doors found in malls that allow a person to enter a room upon detecting motion all the way up to the autonomous vehicles that seemed like a distant future a few decades back. We can make our homes smarter by replacing conventional switches with Passive Infrared (PIR) sensors that can detect motion and intelligently turn the devices in a room on or off based on the presence of people in the room.

To conclude, this project was implemented in an attempt to demonstrate a Home Automation System by using PIR sensors in order to automate certain repetitive tasks in our daily lives, thus saving time and energy.

FUTURE SCOPE

The home automation market is primarily driven by growing need for effective solutions in various domestic applications such as lighting, heating, ventilation, air conditioning, safety and security as well as energy management. This project can be broadened in order to satisfy all of the above use cases. A number of such PIR sensors can be set up in every room of the house. These sensors can then be linked to all of the devices in that particular room.

A system can be designed wherein the PIR sensors in each room turn on a different set of devices based on the user's preference. This type of an approach on Home Automation is particularly helpful for elderly people or people with disabilities.

The project can further be expanded to workplaces such as corporate offices, educational institutions, hospitals and other places in which the energy usage is very large.

REFERENCES

1.) J Bhasker – A Verilog HDL Primer, second edition.

2.) Simon Monk – Programming FPGAs: Getting Started with Verilog.

3.) Basys 3 FPGA Board Reference Manual - https://reference.digilentinc.com/_media/basys3:basys3_rm.pdf

4.) HC – SR501 PIR Sensor Datasheet –

https://components101.com/sensors/hc-sr501-pir-sensor

5.) An overview on Vivado - https://www.xilinx.com/products/design-tools/vivado.html

APPENDIX

Verilog Code

```
module minip(CLK,CLR,PIR_SENSOR_1,PIR_SENSOR_2,RELAY_A,RELAY_B,OV_Enable,OV1,OV2);

output RELAY_A, RELAY_B;

input CLK,CLR,PIR_SENSOR_1,PIR_SENSOR_2,OV_Enable,OV1,OV2;

reg kktemp1,kktemp2;


always@(posedge CLK)
    begin
// SENSOR SWITCH
        if(OV_Enable == 0)
        begin
            if (CLR==1'b1)
            begin
               kktemp1 <= 1'b0;
               kktemp2 <= 1'b0;
            end
            else if ((PIR_SENSOR_1==1'b1) && (PIR_SENSOR_2==1'b0))
            begin
               kktemp1 <= 1'b0;
               kktemp2 <= 1'b0;
            end
            else if ((PIR_SENSOR_1==1'b0) && (
PIR_SENSOR_2==1'b1))
            begin
               kktemp1 <= 1'b1;
               kktemp2 <= 1'b1;
            end
        end
// MANUAL SWITCH
        else
        begin
            if(OV1 == 1 )
                kktemp1 <= 1'b1;
            else
                kktemp1 <= 1'b0;
            if(OV2 == 1 )
                kktemp2 <= 1'b1;
            else
                kktemp2 <= 1'b0;
        end
```

```
    end
    assign RELAY_A=kktemp1;
    assign RELAY_B=kktemp2;
endmodule
```

Constraint File

```
set_property PACKAGE_PIN W5 [get_ports CLK]
set_property PACKAGE_PIN W15 [get_ports CLR]
set_property PACKAGE_PIN R18 [get_ports PIR_SENSOR_1]
set_property PACKAGE_PIN M19 [get_ports PIR_SENSOR_2]
set_property PACKAGE_PIN V3 [get_ports RELAY_A]
set_property PACKAGE_PIN V14 [get_ports RELAY_B]
set_property PACKAGE_PIN W2 [get_ports OV_Enable]
set_property PACKAGE_PIN T1 [get_ports OV1]
set_property PACKAGE_PIN U1 [get_ports OV2]
set_property IOSTANDARD LVCMOS33 [get_ports CLK]
set_property IOSTANDARD LVCMOS33 [get_ports CLR]
set_property IOSTANDARD LVCMOS33 [get_ports PIR_SENSOR_1]
set_property IOSTANDARD LVCMOS33 [get_ports PIR_SENSOR_2]
set_property IOSTANDARD LVCMOS33 [get_ports RELAY_A]
set_property IOSTANDARD LVCMOS33 [get_ports RELAY_B]
set_property IOSTANDARD LVCMOS33 [get_ports OV_Enable]
set_property IOSTANDARD LVCMOS33 [get_ports OV1]
set_property IOSTANDARD LVCMOS33 [get_ports OV2]
```