HDL IMPLEMENTATION OF AUTOMATED RESIDENCE MONITORING AND CONTROL SYSTEM

M.Devadas, M.Bhavani, V.Kavyasri, N.Nikshitha, MP.Kavyasri
Department of Electronics and Communication Engineering, Vaagdevi College of Engineering, Warangal, India

Abstract— Home automation is a system designed to intelligently control electrical and electronic devices in homes, offices, or buildings, prioritizing two key parameters: security and comfort. It also provides the comfort facility of lighting and temperature control. The system features fire detection and intrusion monitoring through doors, windows and garages for security, and provides a foundation for a smart home implementation based on digital systems design principles. It manages both the security and comfort of a house, by way of interfacing devices and sensors. Using a Verilog code; functionality is achieved and the code is synthesized and tested on the Xilinx platform. They are based on the concepts of Finite State Machine (FSM) in an attempt to achieve efficient system operation. The design is validated by simulated waveforms showing the successful implementation.

Keywords—FPGA, Xilinx, Moore Machine, Home Automation System, State diagram, Simulation, Synthesis, verilog HDL

I INTRODUCTION

In today's fast-paced world, a Smart Home system addresses the need for convenience and security by integrating two essential modules: security and comfort. The comfort module seeks to create a pleasant environment; the security module however prevents or helps to identify in intrusions and accidents. This is a firmware based home automation system using the Verilog code such that a password protected module is activated after the arrival of the homeowner. Motion sensors in the home trigger the comfort module, once they give the permission. If the house is unoccupied only the security module is still on to secure the house. The system has multiple sensors interfaced as inputs and is designed and verified through the Xilinx platform to provide reliable and efficient implementation for today's living.

II METHODOLOGY

A. BLOCK DIAGRAM

A comprehensive block diagram comprising of all modules is shown to explain how conditions and states are managed by the system. At its core is the main module, which is divided into two sub-modules: The password module, the comfort module. For homeowner access there is an 11 bit password which the password module can use to activate or deactivate the security system.

Data acquired from interfaced sensors on inputs such as room_door, room_window, garage_door and fire is gathered by the security module. The door magnet module monitors the door status continuously and will set off an open door alarm when its magnet fails and the door is left open. The garage and window modules are like the other. The smoke module activates the fire alarm if the smoke detector senses smoke, setting the fire state to high; otherwise and the fire state remains low.

Upon entering the correct password, the security module is activated, subdividing further into door, window, garage, and smoke modules. Meanwhile, the comfort module manages the home's environment through temperature and luminosity controllers. The system includes four types of alarms: door, window, garage, and fire alarms, ensuring comprehensive monitoring and response to various conditions..

Fig 1 shows the block diagram of Automated Residence monitoring and control with Verilog

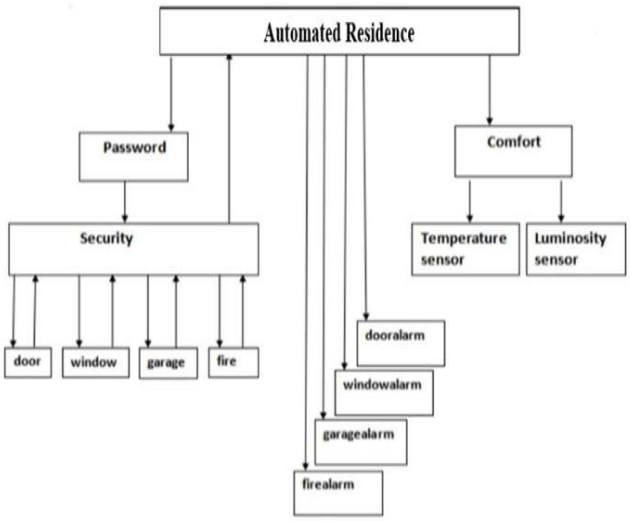

Fig 1: Block Diagram of Automated Residence

Fig 2 illustrates a flowchart for a home automation system that manages temperature, lighting, and security. The process begins at the "Start" node, where the system monitors the temperature using a temperature sensor. If the temperature is below the desired level, the heater is activated; if it is above, the cooler is turned on. After adjusting the temperature, the system returns to a "Normal" state and continues monitoring.

A password secures an access to the system. The process will run if the correct password is entered, but if not, it will trigger an alarm. The lighting also changes according to an input from a light sensor. If the light level is high, the lights glow brighter; if low, the lights dim. After adjustment, the system resets and continues monitoring in "Normal" mode.

Another is the ability to detect smoke. When smoke is detected an alarm is triggered and the system switches state back to “Alarm off.” The system checks if the door is open for door status. The guy, if its his system turns the lights on. If closed, the system goes back to "normal." The last node of the process is called End.

The significant systems in this home automation flowchart are actually controlling home's temperature, lighting, and security by sensing and meeting a condition, dynamically adjusting based on sensor's input and the predefined condition of the home.

B. FLOW CHART

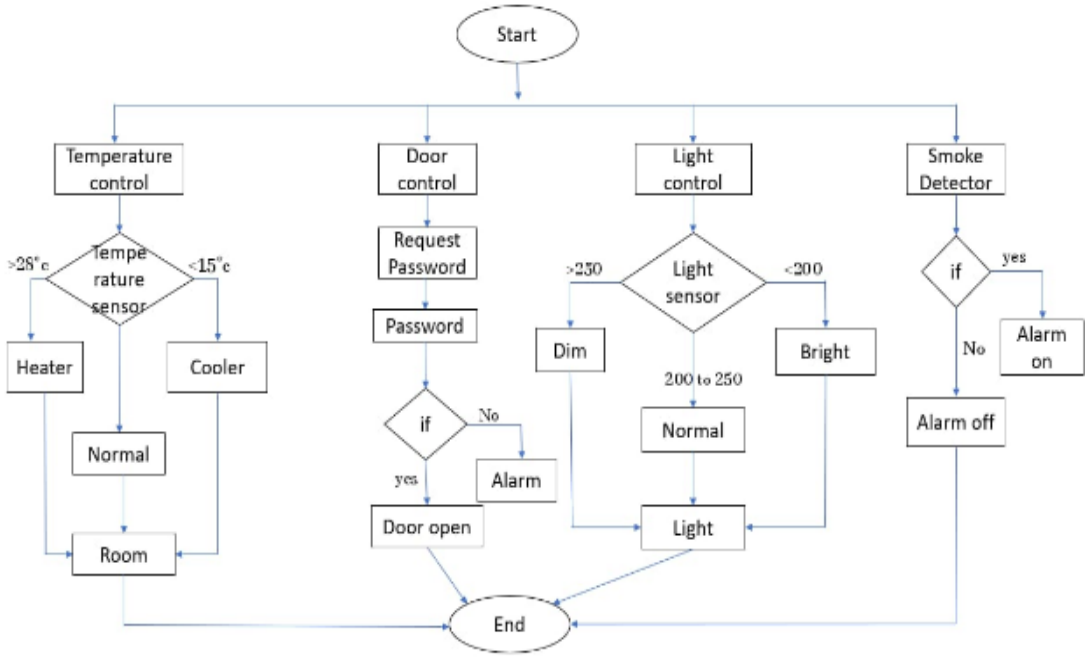

Fig 2: Flow chart of Automated Residence

C. STATE DIAGRAM

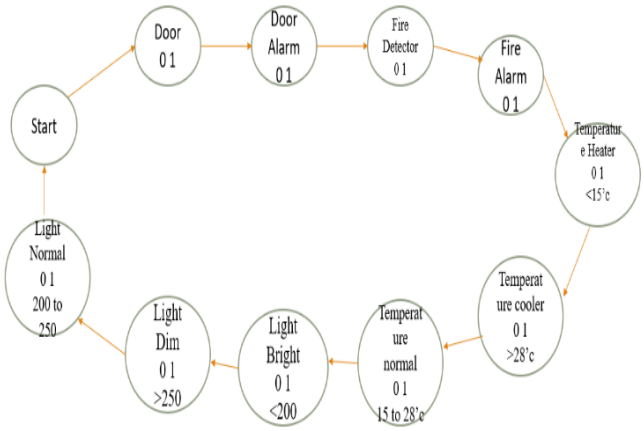

Fig 3: State Diagram of Automated Residence

The system for monitoring and controlling the environment of a room is symbolized by this state diagram. Here is an overview of its states and transitions:

The System begins in Start state. When the door is opened, the door changes state to "Door" meaning someone is entering or exiting this room. If for an extended period the door is open you will be in "Door Alarm." In 'Door' state, when the door is closed the system changes back to' Door' state. The system goes to "Fire Detector" state when a fire is found while in state "Door" and when fire alarm discharges, the state become "Fire Alarm". When the temperature drops below 15°C, the system goes to a 'Temperature Heater' state after the fire alarm is deactivated.

The system also includes other temperature-related states: Once the temperature reaches above 28°C, “Temperature Cooler” is active and if the temperature stays between 15°C and 28°C, “Temperature Normal” is maintained.

D. Window control

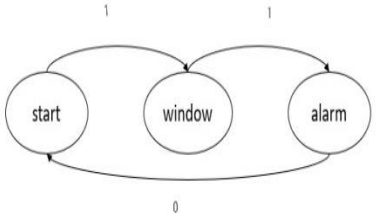

Fig 4: window control module

The image depicts a basic state diagram with three states: It includes ‘Start‘, ‘Window’ and ‘Alarm’. It starts from "Start", goes to "Window", "Alarm", etc. There is a loop back to the "Start" from the "Alarm" state... meaning the sequence can (and does) repeat.

E. Temperature control

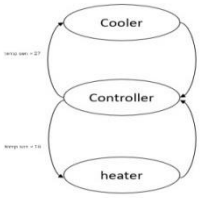

Fig 5: Temperature Module

The system is designed to operate based on the people present in the hall and the temperature. If the hall temperature is reaches to below the 18°C, the system turn on the heater stage and activates the heater until the hall temperature reaches to 21°C. During this period, it remains in the Controller Stage. In the Heater Stage, a counter is used to manage operations, with the heater turning on as required.

F. Door control

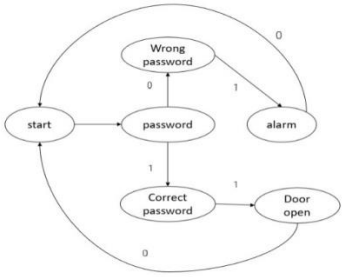

Fig 6: Door control module

This system ensures that only the authorized person can enter by verifying the user ID and password. The process begins when the entrance sensor detects the presence of an individual. If motion is detected, the system proceeds to the password checking stage. If the entered password matches the stored credentials, the system moves to the **correct password** stage. Once a person enters, the entrance sensor rechecks for presence. If no motion is detected, the system returns to the **start stage**. If the password is incorrect, the system will continue checking until the correct password is entered. The system's status is indicated by **green** and **red** LED lights: a red light blinks for an incorrect password, while a **green** light is shown for the correct one. If no action is detected, no light will blink, indicating the idle state. The flow of this security check module is depicted in fig 6.

G. Light control

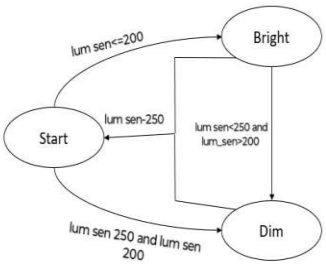

Fig 7: Light Module

The luminous sensor, lum_sen, is used to detect light levels. When the sensor detects a value less than or equal to 200, the system transitions to the "bright" state. If the detected value falls between 200 and 250, the system enters the "dim" state. This procedure is activated only when the count of people is greater than 0. If no people are detected or until this condition is met, the system remains in the "startl" state. This module is illustrated in Fig 7.

III RESULTS AND DISCUSSION

A Verilog implementation of the system would involve creating a module to represent the entire system, with input and output declarations for signals such as clk, password, garage_door, and alarm. The internal logic would utilize Verilog constructs like if, else, case statements, and combinational logic gates to manage password verification, control the garage door, and activate the alarm. Synchronization with the clk signal would ensure that all operations occur at the appropriate times. A state machine could be employed to manage various system states, such as idle, waiting for password input, opening the garage door, or triggering the alarm. To ensure system security, robust measures would be implemented to prevent unauthorized access and password cracking.

A. Light Control

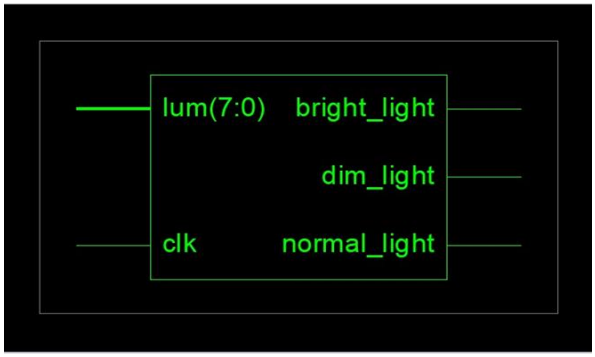

Fig 8: RTL Schematic view of Light Control

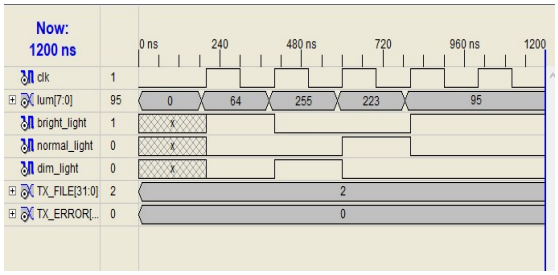

Fig 9: Simulated Waveform of Light Control

The clk signal is a binary clock that oscillates between 0 and 1 with a consistent period of 240 ns, providing synchronization and timing for the system. The waveform shows rising edges at 0, 240, 480, 720, and 960 ns, marking the regular clock cycle. The lum[7:0] signal, which represents luminance levels, starts at 0 ns and gradually increases, reaching 64 at 240 ns, peaking at 255 at 480 ns, and then decreasing to 223 at 720 ns and 95 at 960 ns, before stabilizing at 95 by 1200 ns.

The bright_light signal is active between 240 ns and 720 ns, corresponding to high luminance levels (64, 255, and 223), while the normal_light signal is active from 0 to 240 ns and from 960 ns onward, indicating moderate luminance levels (0, 64, and 95). The dim_light signal remains inactive throughout, indicating no low-luminance conditions.

The TX_FILE [31:0] output signal, representing data transmission, starts at 0 and transitions to 2 at 960 ns, signaling the initiation of file transmission based on system

triggers. Meanwhile, the TX_ERROR signal remains inactive, indicating that no errors occur during the process. The system effectively adjusts light levels based on luminance inputs, with synchronization provided by the clock signal, and manages data transmission without errors.

B. **Door Control**

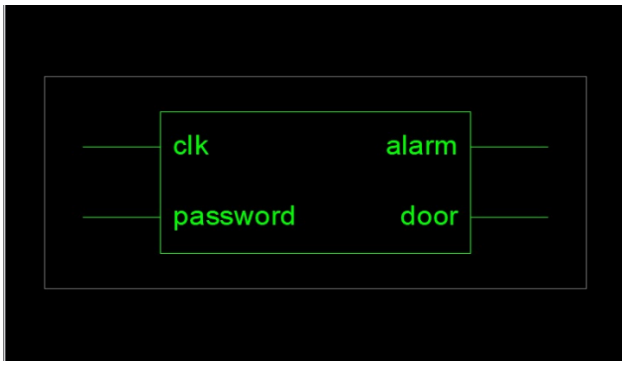

Fig 10: RTL schematic view of Door Control

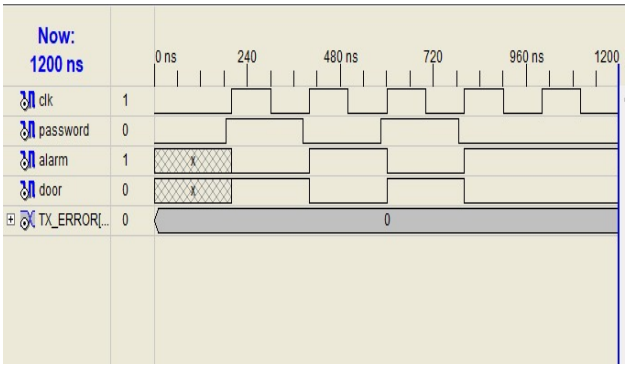

Fig 11: Simulated Waveform of Door Control

The system's operation is driven by the clk signal, a periodic square wave that serves as the timing reference for the entire circuit. The password signal represents input from a password entry device, going high (logic 1) for correct entries and low (logic 0) otherwise. The alarm signal activates (goes high) upon detecting an incorrect password or an unauthorized attempt, while the door signal controls the door state, going high to open and low to close it.

The password signal is sampled on the rising edge of the clock, allowing the system to verify the entered password at each clock cycle. If the password is correct, the door signal changes state to grant access. If the password is incorrect, the alarm is triggered. The TX_ERROR signal, indicating transmission errors, remains low throughout the observed period, confirming no errors occurred.

The system operates with the clock providing synchronization, while the password input is verified against a stored value. The door or alarm outputs are then activated accordingly. Internally, password verification likely involves comparing the input to a stored correct password using logic gates or a comparator. If the password matches, the system grants access by controlling the door output; otherwise, the alarm is triggered. Timing is carefully managed through the clk signal to ensure synchronization for verification and control.

C. **Temperature Control**

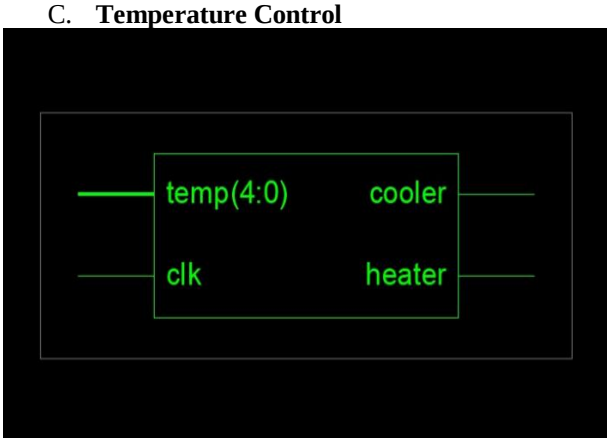

Fig 12: RTL Schematic view of Temperature Control

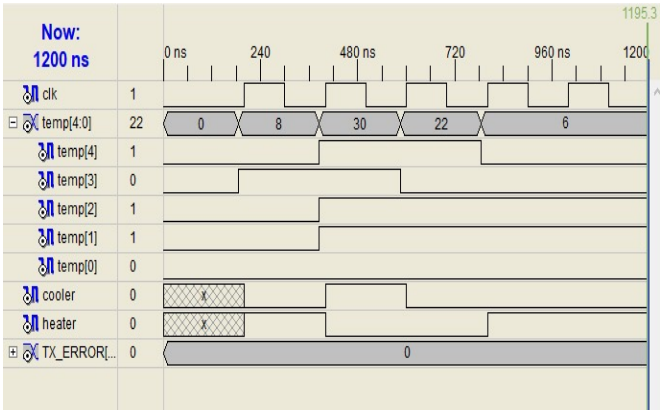

Fig13: Simulated waveform of Temperature Control

The simulated waveform of the temperature control system demonstrates the operation of the clk signal, which toggles consistently between logic 1 and 0 with a period of 240 ns. This clock signal governs the timing, controlling the states and transitions of other signals. The temp[4:0] signal, a 5-bit representation of the temperature, indicates the system's current temperature readings. At various points, the temp values are recorded as 0 at 0 ns, 8 at 240 ns, 30 at 480 ns, 22 at 720 ns, and 6 at 960 ns, all synchronized with the clock. The temp[4] signal represents the most significant bit (MSB), and temp[0] represents the least significant bit (LSB).

D. **Window Control**

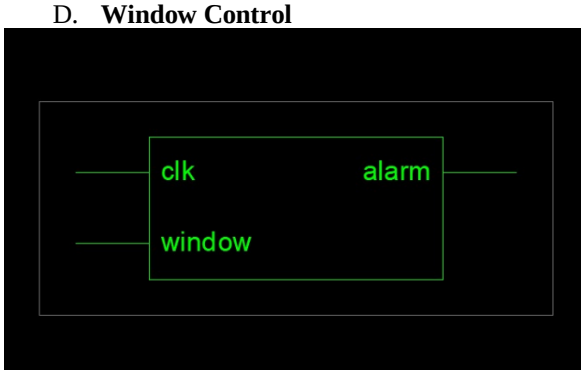

Fig 14: RTL Schematic View of Window Control

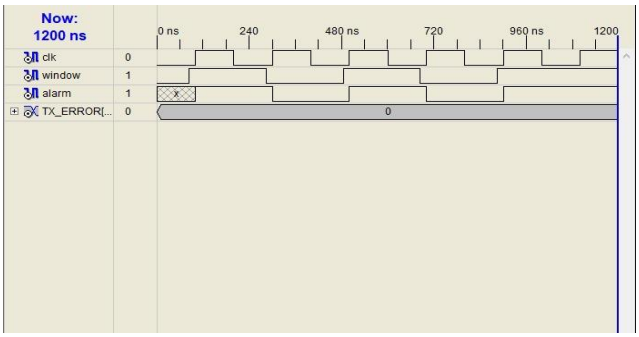

Fig 15: Simulated Waveform of Window Control

The clk signal is a periodic square wave that acts as the system clock, providing the timing reference for all operations. The window signal controls the state of the window, going high to open it and low to close it. The alarm signal serves to indicate errors or unauthorized access, going high when such conditions occur. The TX_ERROR signal, which flags transmission errors, remains low throughout the observation period, signifying that no transmission issues are present.

The window signal is sampled on the rising edge of the clock, meaning the system checks the window's state in sync with the clock cycle. However, the alarm signal stays high for the duration of the observation, indicating a persistent error or unauthorized access. During this period, the window signal remains low, suggesting that the system is unable to control the window.

E. **Smoke Detector**

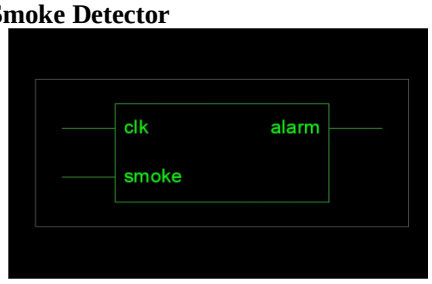

Fig 16: RTL Schematic View of Smoke Detector

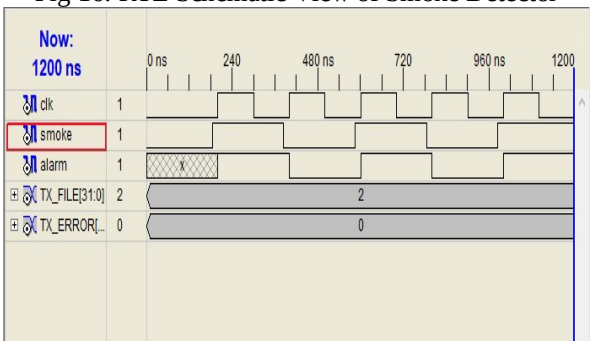

Fig 17: Simulated Waveform of Smoke Detector

The clk signal is a periodic square wave serving as the system clock, providing a stable timing reference for the entire circuit. The smoke signal, which is the output of a smoke detector, goes high (logic 1) when smoke is detected and low (logic 0) when no smoke is present. The alarm signal activates (goes high) when the smoke signal remains high for a sustained period, indicating a potential fire hazard. The TX_FILE[31:0] signal is a 32-bit data bus transmitting information related to the smoke detection system, with its changing values reflecting ongoing data transmission.

The TX_ERROR signal flags transmission errors but remains low throughout the observation period, indicating that the data transfer is reliable. The smoke signal is continuously sampled at each clock cycle, while the alarm signal is triggered only after the smoke signal stays high for multiple clock cycles. Meanwhile, the system efficiently transmits data about smoke detection and alarm status via the TX_FILE bus, without any transmission errors, as indicated by the low TX_ERROR signal.

In summary, the system relies on the clock for synchronization, detects smoke via the smoke input, activates the alarm upon sustained smoke detection, and transmits related data efficiently.

F. **Garage Control**

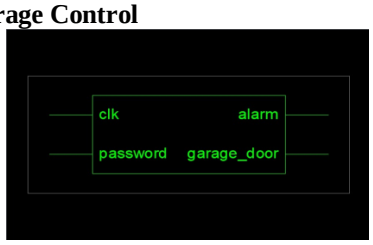

Fig 18: RTL Schematic View of Garage Control

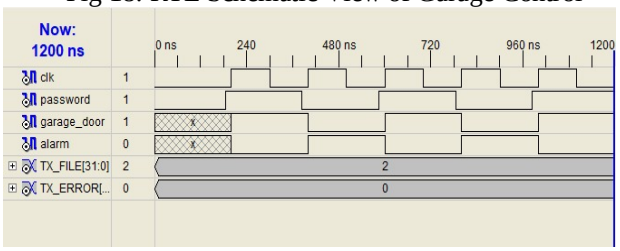

Fig 19: Simulated Waveform of Garage control

The clk signal, a periodic square wave, serves as the system clock, providing consistent timing and synchronization for the entire circuit. The password signal represents input from a password entry device, going high (logic 1) when the correct password is entered and low (logic 0) otherwise. The garage_door signal is an output used to control the garage door, where it goes high to open the door and low to close it. The alarm signal serves as an error indicator, activating when an incorrect password is entered, signaling unauthorized access. The TX_FILE[31:0] signal is a 32-bit data bus that transmits data related to the garage door system, with its varying values indicating active data transmission. The TX_ERROR signal flags transmission issues but remains low throughout, indicating reliable data transfer without errors.

The password signal is sampled on the rising edge of the clk signal, allowing the system to verify the entered password during each clock cycle. If the correct password is entered, the garage_door signal changes state, controlling the door accordingly. If the password is incorrect, the alarm signal activates to indicate unauthorized access. Meanwhile, the TX_FILE bus transmits data related to the password authentication and garage door operations, and the TX_ERROR signal remains low, confirming error-free transmission.

In summary, the system operates as follows: the clock provides synchronization, the user inputs a password, which is verified against a stored value. If the password is correct, the garage door operates; if incorrect, the alarm is activated. Data related to these operations is transmitted reliably via the TX_FILE bus, with error-free transmission indicated by the low TX_ERROR signal.

The output waveform, as shown in Fig. 19, displays the values of various signals over time. Time is represented along the top axis, with signal values in gray boxes. The clk signal toggles every 240 ns to synchronize the other signals. Boolean signals such as window, door, and smoke_alarm indicate the status of different system components. The lum[7:0] and temp[4:0] signals represent numerical values, while TX_FILE[31:0] and TX_ERROR[31:0] handle data transmission and error management. Signals change in sync with the clock or after delays of 480 ns, 720 ns, or 960 ns, depending on the system operation.

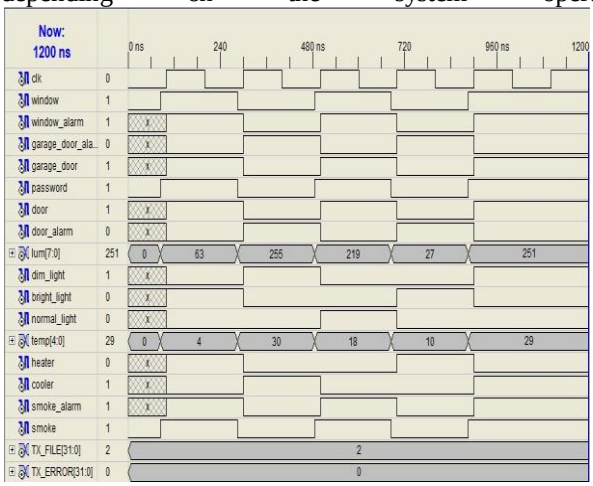

Fig 20:Xilinx output waveform after simulation for Automated Residence

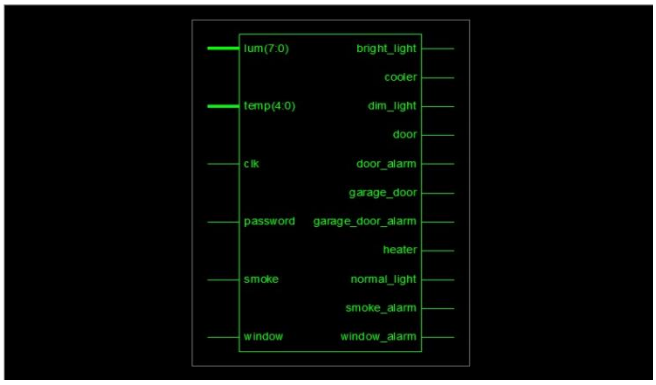

Fig 21: RTL Schematic view of design

IV CONCLUSION

This paper formally introduces a cost effective automated home security system. There were several assumptions made throughout the development process but great effort was taken to ensure that the practices itself are practical. The system checks the devices in a prioritized order and the display displays the system status. The system presents itself as a low cost and efficient solution that could be further developed, making it web enabled for other functionality. It is a straightforward approach to implement and can be trivially integrated into other devices.

V REFERENCES

[1]. M. S. Ahmed, R. Mukherjee, P. Ghosh, S. Nayemuzzaman and P. Sundaravdivel, "FPGA-based assistive framework for smart home automation," *2022 IEEE 15th Dallas Circuit And System Conference (DCAS)*, Dallas, TX, USA, 2022, pp. 1-2,

[2] R. Payal, A. Saxena and B. Chanda, "Implementation of Smart Home through FPGA using Verilog Hardware Descriptive Language," *2020 IEEE International Conference on Advent Trends in Multidisciplinary Research and Innovation (ICATMRI)*, Buldhana, India, 2020, pp. 1-6,

[3]. S. Sen, S. Saha and A. Dasgupta, "HDL-Based Smart Home Implementation Using FSM Logic and FPGA," *2023 11th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON)*, Jaipur, India, 2023, pp. 1-5

[4] S. Sai Vandana, K. Abinandhan and S. R. Ramesh, "Implementation of Smart Home using Finite State Machine Model," *2022 7th International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, 2022, pp. 45-49,

[5]. Komol Arafat. Gani, Farzana. Yasmin, A B M Najmul. Karim, Iqbalur. Rahman .” Home Automation System Design Using Verilog Hardware Descriptive Language” , 2nd International Conference on Recent Trends in Computer and Information Engineering (ICRTCIE'2013) Dec. 20-21, 2013 Bali (Indonesia)

[6]. S. Sharma, J. Boddu, G. S. Charan, S. Sharma, S. Sivanantham and K. Sivasankaran, "Home automation through FPGA controller," *2015 Online International Conference on Green Engineering and Technologies (IC-GET)*, Coimbatore, India, 2015, pp. 1-4,

[7]. Prof. Dr. Sanjeev Sharma, Ms. Revati Deokar” FPGA Based Cost Effective Smart Home Systems “ , International Conference On Advances in Communication and Computing Technology , February 2018.

[8] N.Chintaiah, K. Rajasekhar, V. Dhanraj, “ Automated Advanced Industrial and Home Security using GSM and FPGA” , International Journal of Computer Science and Information Technologies, vol. 2(4), 2011, 1598- 1602.

[9]Carl J. Debono and Kurt Abela Department of Communications and Computer Engineering “Implementation of a Home Automation System through a Central FPGA Controller “, University of Malta, January 2018.

[10] K. Madhuri, B. L. Sai and B. S. Sirisha, “A home automation system using hardware descriptive tool,” Ijert, vol.2, July 2013.

[11] Palnitkar, S. (2008). Verilog HDL (2nded.). Pearson Education.

[12] Floyd, “Electronic Devices”, Pearson Education International, 2005.