HOME AUTOMATION AND SECURITY SYSTEM

CSE–4224: Digital System Design Laboratory

Submitted By

Israt Tasnim Esha
Roll: 1907090

Submitted To:

Md. Badiuzzaman Shuvo

Lecturer

Dept. of Computer Science and Engineering

Khulna University of Engineering & Technology

Md. Sakhawat Hossain

Lecturer

Dept. of Computer Science and Engineering

Khulna University of Engineering & Technology

Objectives:

- To design and implement a **Verilog-based Home Security System** integrating multiple subsystems for enhanced safety and energy efficiency.
- To develop a system that incorporates **password protection**, ensuring secure access to the premises.
- To implement **fire detection mechanisms** using sensors for temperature and smoke levels, enabling immediate hazard alerts.
- To create an **automated door control system** that ensures safety by managing door operations within a predefined time limit.
- To implement **human presence detection** for controlling room automation, such as AC and light, to enhance comfort and save energy.
- To test and validate the system design through simulation, ensuring functionality and robustness of the proposed digital logic system.

Project Description:

The project focuses on the design and implementation of a Verilog-based Home Security and Automation System that combines multiple safety and energy-saving features. This system integrates password protection, fire detection, and automated room management to enhance the security, comfort, and efficiency of a home or building.

Features of the System

The Verilog-based Home Security and Automation System integrates several features to ensure safety, automation, and energy efficiency. Each feature has been designed and implemented with a focus on reliability and functionality. The detailed features are as follows:

1. Password Protection and Security

- Password Storage: The system stores a secure 4-digit password in Read-Only Memory (ROM).
- User Authentication: Users must enter the correct password to gain access. Upon successful entry, the door opens automatically.
- Incorrect Password Handling:
  - If the user enters an incorrect password, the system denies access and raises an alert.
  - To prevent brute-force attacks, there is a retry limit set to 3 attempts. After three consecutive incorrect attempts, the system triggers a security alarm.

2. Fire Detection

- Smoke and Temperature Monitoring: The system continuously monitors the smoke level and ambient temperature using sensors.
- Alarm Activation: If smoke or temperature exceeds a predefined threshold:

    - The fire alarm is triggered instantly to alert the inhabitants of potential danger.
- Real-Time Monitoring: The sensors work in real-time, ensuring immediate response to hazardous conditions.

3. Door Control System

- Timed Door Operation: The system automates door control with the following functionalities:
    - The door opens automatically upon password authentication.
    - If the door remains open for more than 2 minutes, it closes automatically for safety.
- Manual Override: The door can also be operated manually, but the system continues to monitor the time to maintain security.

4. Human Presence Detection

- Automation Based on Human Presence: The system utilizes sensors to detect human presence in the room and adjusts environmental settings accordingly:
    - If a person is detected and the temperature exceeds 30°C, the system turns on the air conditioner (AC) and lights to ensure comfort.
    - If no human presence is detected, the AC and lights are turned off to conserve energy.
- Energy Efficiency: This feature ensures optimal use of energy by operating appliances only when necessary.

5. Finite State Machine (FSM) Control

- State-Based Management: The entire system operates using a finite state machine (FSM) that transitions through various states like:
    - Password Entry
    - Door Control
    - Fire Detection
    - Human Presence Monitoring
- Efficient Workflow: The FSM ensures a systematic and organized flow of operations, reducing system complexity and improving reliability.

6. Simulation and Testing

- The system is designed to be tested and validated using Active HDL simulation software.
- Each feature, such as password validation, fire detection, and human presence automation, is individually tested to ensure functionality and robustness.

	pm	ptl	dtl	tl	pt	pp	rh	p	f	d		D3	D2	D1	D0
t0	X	X	X	X	X	X	X	1	X	X	t1	0	0	0	1
t0	X	X	X	X	X	X	X	X	1	X	t8	1	0	0	0
t0	X	X	X	X	X	X	X	X	X	1	t10	1	0	1	0
t1	X	X	X	X	X	X	X	X	X	X	t2	0	0	1	0
t2	0	0	X	X	X	X	X	X	X	X	t2	0	0	1	0
t2	X	1	X	X	X	X	X	X	X	X	t3	0	0	1	1
t2	1	X	X	X	X	X	X	X	X	X	t5	0	1	0	1
t3	X	X	X	X	X	X	X	X	X	X	t4	0	1	0	0
t5	X	X	X	X	X	X	X	X	X	X	t6	0	1	1	0
t6	X	X	0	X	X	X	X	X	X	X	t6	0	1	1	0
t6	X	X	1	X	X	X	X	X	X	X	t7	0	1	1	1
t7	X	X	X	X	X	X	X	X	X	X	t4	0	1	0	0
t8	X	X	X	X	0	X	X	X	X	X	t8	1	0	0	0
t8	X	X	X	X	1	X	X	X	X	X	t9	1	0	0	1
t9	X	X	X	X	X	X	X	X	X	X	t4	0	1	0	0
t10	X	X	X	X	X	0	X	X	X	X	t10	1	0	1	0
t10	X	X	X	0	X	X	X	X	X	X	t10	1	0	1	0
t10	X	X	X	1	X	1	X	X	X	X	t11	1	0	1	1
t11	X	X	X	X	X	X	X	X	X	X	t12	1	1	0	0
t12	X	X	X	X	X	X	0	X	X	X	t12	1	1	0	0
t12	X	X	X	X	X	X	1	X	X	X	t13	1	1	0	1
t13	X	X	X	X	X	X	X	X	X	X	t4	0	1	0	0
t4	X	X	X	X	X	X	X	X	X	X	t0	0	0	0	0

Here are 4 equations derived from the provided Verilog code:

D3 Equation:

- D3 = (load && (t0&f | t0&d | t8&~pt | t8&pt | t10&~pp | t10&~tl | t10&tl&pp | t11 | t12&~rh | t12&rh));

D2 Equation:

- D2= (load && (t2&pm | t3 | t5 | t6&~dtl | t6&dtl | t7 | t9 | t11 | t12&~rh | t12&rh | t13));

D1 Equation:

- D1= (load && (t0&d | t1 | t2&~pm&~ptl | t2&ptl | t5 | t6&~dtl | t6&dtl | t10&~pp | t10&~tl | t10&tl&pp));

D0 Equation:

- D0= (load && (t0&p | t2&ptl | t2&pm | t6&dtl | t8&pt | t10&tl&pp | t12&rh));

Pseudocode for the System:

Main Control Module (control)

1. **Inputs**:
   - Clock (clk), Reset (reset), Load (load)
   - System Flags: p, f, d, pm, ptl, dtl, tl, pt, pp, rh
   - State outputs: t0, t1, ..., t13
2. **State Transition Logic**:
   - Calculate next state bits (D3, D2, D1, D0) using logical expressions based on current state and inputs:
     - D3: Includes conditions for fire alarms, door locks, and AC operations.
     - D2: Includes password management, state updates, and room management.
     - D1: Includes door open checks, incorrect password, and time limit conditions.
     - D0: Includes password match, fire detection, and environmental conditions.
   - Use a 4-bit register to store the next state.
3. **State Update**:
   - Based on the 4-bit state value, activate corresponding outputs (t0, t1, ..., t13) using a case statement.
4. **Reset and Initialize**:
   - When reset is active, reset all states to initial (t0 = 1).
   -

Breadboard Module

1. **Inputs**:
   - Simulation inputs: clk, reset, load
   - Flags for password, fire, and AC (p, f, d, pm, ptl, dtl, tl, pt, pp, rh)
2. **Memory Initialization**:
   - Initialize ROM addresses for password and fire threshold.
   - Set initial values for variables (count, en, etc.).
3. **Simulation Setup**:

- Configure initial inputs and states (p, f, d, etc.).
- Reset the system.
- Enable the main system (load = 1).

4. **Password Management**:
   - If password is correct, allow transition to the next state.
   - If incorrect and exceeds retry limit, trigger alarm.
5. **Fire Detection**:
   - Monitor fire flags (f, pt) to trigger fire alarms.
   - Reset to initial state after managing fire conditions.
6. **AC Control**:
   - If room conditions match (pp, tl, rh), toggle AC ON/OFF states.
   - Use conditions for human presence and temperature to manage transitions.

Half Adder Module

1. **Inputs**:
   - Two binary inputs (a, b)
2. **Outputs**:
   - Sum (s), Carry (c)
3. **Logic**:
   - s = a XOR b
   - c = a AND b

Full Adder Module

1. **Inputs**:
   - Two binary inputs (a, b) and carry-in (cin)
2. **Outputs**:
   - Sum (s), Carry-out (cout)
3. **Logic**:
   - Use two half-adders:
     - Half-adder 1: Compute sum (s1) and carry (c1) of a and b.

        - Half-adder 2: Compute final sum (s) and carry (c2) using s1 and cin.
    - Compute cout = c1 OR c2.

ROM Module

1. **Inputs**:
    - Address (addr)
2. **Outputs**:
    - Data corresponding to the address.
3. **Logic**:
    - Fetch data from the ROM based on input address.

Comparator Module

1. **Inputs**:
    - Two binary values (A, B)
2. **Outputs**:
    - A_greater_B, A_equal_B, A_less_B
3. **Logic**:
    - Compare A and B to generate outputs:
        - A_greater_B = 1 if A > B
        - A_equal_B = 1 if A == B
        - A_less_B = 1 if A < B

Simulation and Monitoring

1. **Simulation Steps**:
    - Simulate different states by toggling clk and load.
    - Test password entry, fire detection, and AC control scenarios.
    - Monitor transitions and validate outputs.
2. **Output Monitoring**:
    - Display state transitions and output values in real-time for verification.

Parallel Adder Module

1. **Inputs**:
   - 4-bit inputs a and b
   - Control signals s0, s1
   - Carry input cin
2. **Outputs**:
   - 4-bit sum s
   - Carry output cout
3. **Logic**:
   - Use 4 cascaded **comb_full_adder** modules to compute:
     - Sum for each bit s[i]
     - Carry for the next bit c[i]
   - Output the final carry as cout.

Comb Full Adder Module

1. **Inputs**:
   - Binary inputs a, b
   - Control signals s0, s1
   - Carry input cin
2. **Outputs**:
   - Sum (s)
   - Carry (cout)
3. **Logic**:
   - Based on s0 and s1, generate intermediate signals p:
     - If s0=1 and s1=0, p = ~b
     - If s0=0 and s1=1, p = b
   - Use a **full adder** to compute:
     - Sum s = a + p + cin
     - Carry cout

ROM Module

1. **Inputs**:
   - Address addr (2 bits)
2. **Outputs**:
   - 4-bit data output data
3. **Logic**:
   - Predefined memory content:
     - Address 0: Password
     - Address 1: Particle Limit
     - Address 2: Temperature Limit
     - Address 3: Room Cool Limit
   - Use a **decoder** to select the memory location.
   - Output the corresponding 4-bit data.

Register Module

1. **Inputs**:
   - 4-bit data D
   - Clock signal clk
   - Reset signal rst
   - Enable signal en
2. **Outputs**:
   - 4-bit data Q
3. **Logic**:
   - Use four **D flip-flops**:
     - On clock edge, if rst=1, reset Q to 0.
     - If en=1, store input D in Q.

Comparator Module

1. **Inputs**:
   - 4-bit values A and B
   - Control signals: clk, rst, en

2. **Outputs**:
    - Signals indicating comparison:
        - A_greater_B (if A > B)
        - A_equal_B (if A == B)
        - A_less_B (if A < B)
3. **Logic**:
    - Use two **registers** to store A and B values.
    - Perform bitwise comparison:
        - If A > B, set A_greater_B = 1
        - If A == B, set A_equal_B = 1
        - If A < B, set A_less_B = 1

D Flip-Flop Module

1. **Inputs**:
    - Data input D
    - Clock signal clk
    - Reset signal rst
    - Enable signal en
2. **Outputs**:
    - Output Q
3. **Logic**:
    - On a clock edge:
        - If rst=1, set Q = 0
        - If en=1, set Q = D

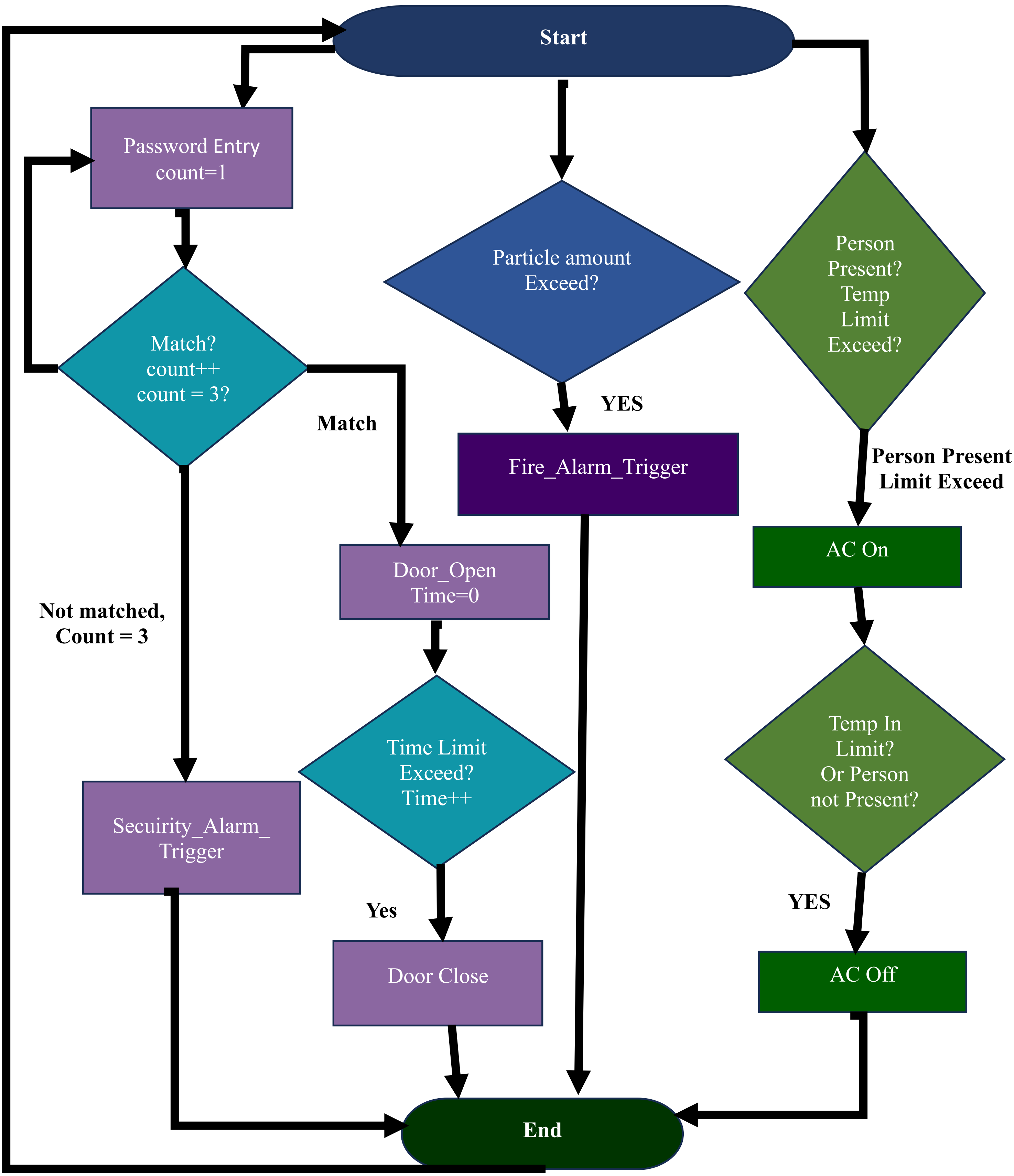
System Overview:

1. **Parallel Adder**:
    - Adds two 4-bit numbers (a and b) based on control signals and generates a 4-bit sum (s) and carry (cout).
2. **ROM**:

- Stores predefined values like passwords, thresholds, and limits.
- Outputs data based on a 2-bit address.

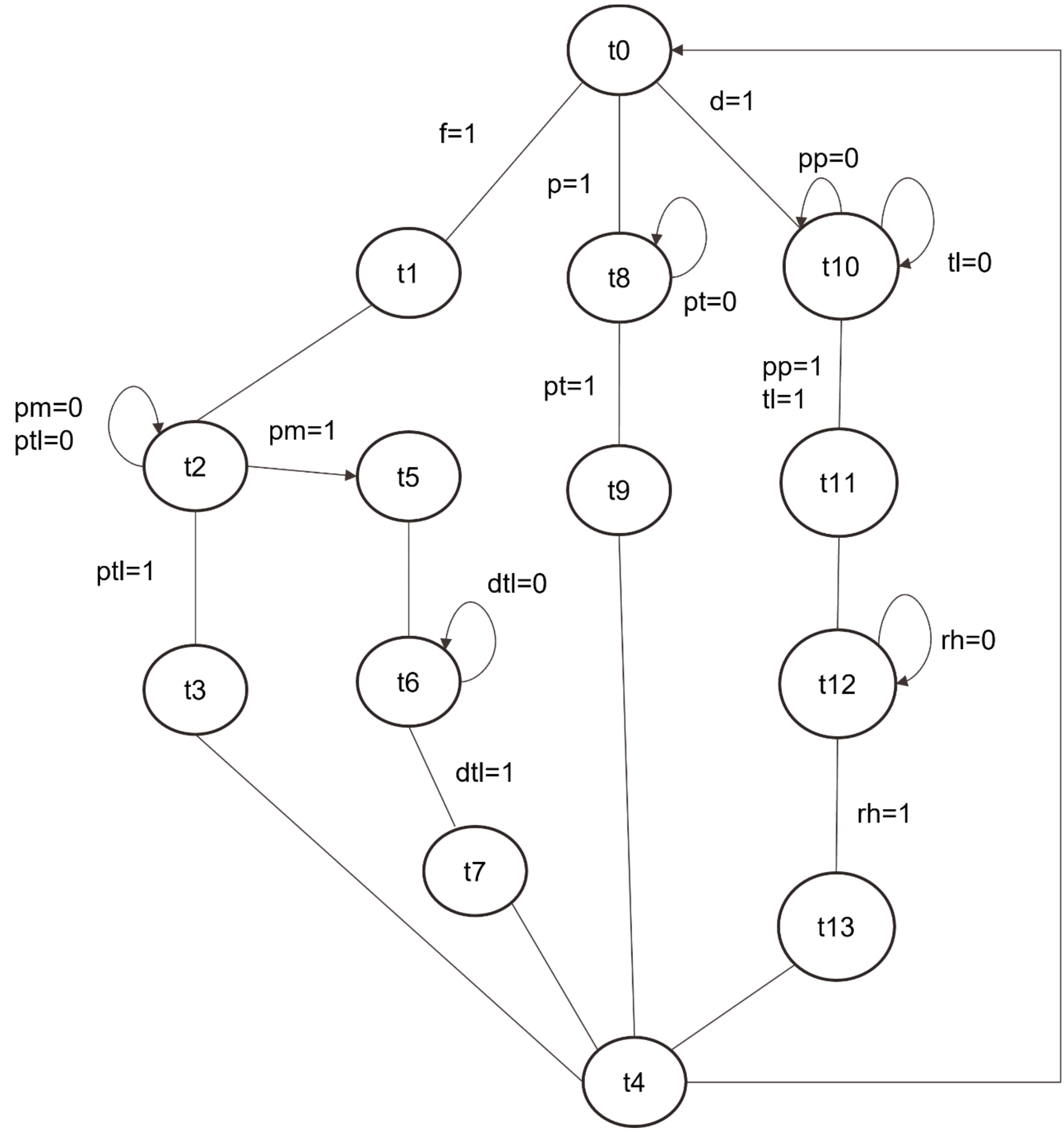
3. **Registers**:
   - Store inputs or outputs for different modules.
   - Update data only when enabled.
4. **Comparator**:
   - Compares two 4-bit numbers (A and B) and outputs the comparison result.
5. **D Flip-Flops**:
   - Building blocks for registers to hold 1-bit data values.
6. **Integration**:
   - These modules work together to create a system capable of performing arithmetic, storage, comparison, and state transitions for the intended application.

Flow Chart



Start
Password Entry
count=1
Match?
count++
count = 3?
Match
Not matched,
Count = 3
Door_Open
Time=0
Time Limit
Exceed?
Time++
Yes
Door Close
Secuirity_Alarm_
Trigger
Particle amount
Exceed?
YES
Fire_Alarm_Trigger
Person
Present?
Temp
Limit
Exceed?
Person Present
Limit Exceed
AC On
Temp In
Limit?
Or Person
not Present?
YES
AC Off
End

State Diagram



Here,

pm = Password Match

ptl = Password Trial Limit

dtl = Door Time Limit

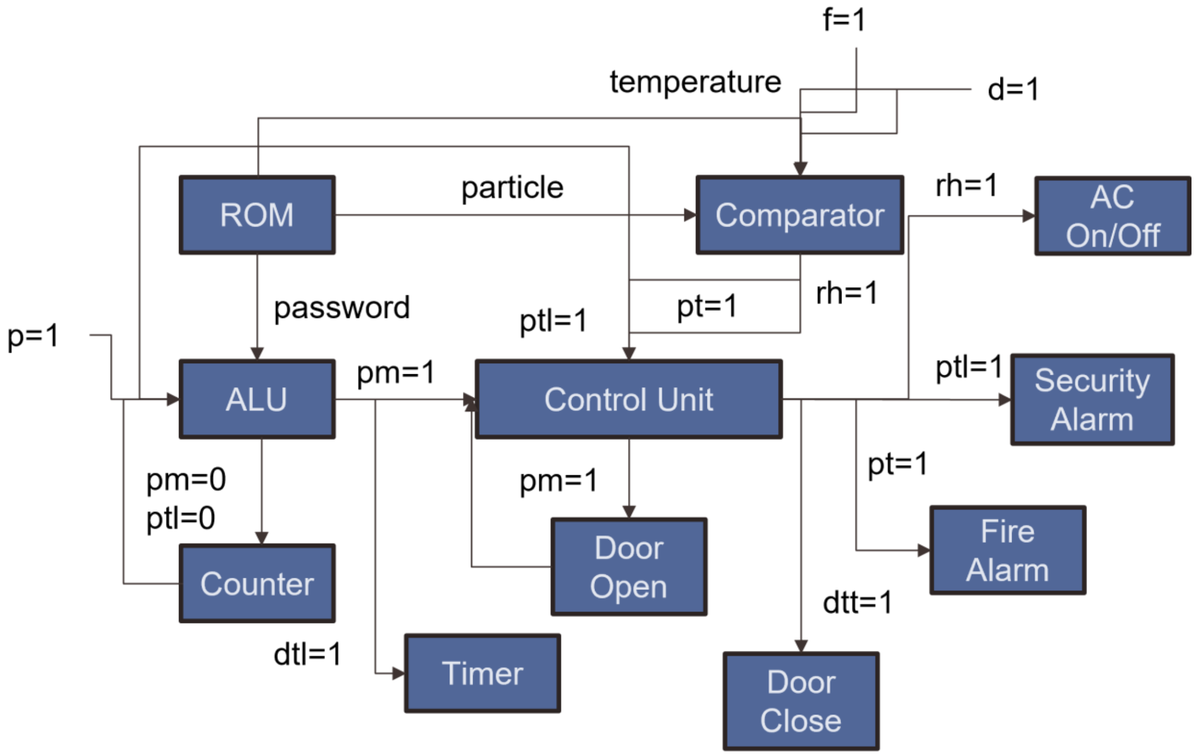
tl = Temperature Limit

pt = Pressure Limit

pp = Person Present

rh = Room Temperature

Block Diagram



Discussion

In this project, we have designed and implemented a Home Security System using Verilog, integrating various safety and automation features. We have used a finite state machine (FSM) to manage operations like password entry, fire detection, and room automation. The system has included password protection with retry limits, fire detection using temperature and smoke sensors, and automated control of doors, lights, and AC based on human presence. We have simulated and tested the design to ensure that all components work correctly and meet the project requirements. This project has provided practical experience in Verilog programming and digital system design.

Conclusion

In conclusion, we have successfully designed and implemented a Verilog-based Home Security and Automation System, integrating key features such as password protection, fire detection, and automated room management. The system has effectively demonstrated secure access control, real-time hazard detection, and energy-efficient automation using a finite state machine (FSM) for reliable operation. Through simulation and testing, we have ensured the system's functionality and robustness. This project has enhanced our understanding of Verilog programming and digital logic design while showcasing its practical applications in modern safety and automation systems.

References

- Lab Slides
- Class Slides