EE 241 Research and Design

Design and Implementation of a Combinational Logic-Based Environmental Control System for Asthma Management in Smart Homes

Sena Melis Tutar

kişi, şahıs, giyim, metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

# Aim of My Home Automation System

To create a home-wide system that monitors and controls environmental factors to help manage asthma and avoid inflammations.

## Monitoring and Control Strategies

1. **Use Air Purifiers**: To reduce allergens and irritants in the air.
2. **Maintain Cleanliness**: Regularly clean and vacuum to reduce dust and pet dander.
3. **Control Humidity**: Use dehumidifiers or humidifiers to keep indoor humidity levels in check.
4. **Avoid Smoking**: Ensure a smoke-free environment.
5. **Monitor Air Quality**: Use air quality sensors and apps to stay informed about outdoor and indoor air quality.

|  |  |  |
| --- | --- | --- |
| System Requirements |  |  |
| Inputs (Sensors) | Outputs (Actuators)  (0: off / 1:on) | Binary Logic Functions |
|  |  | DH=H |
|  | Htr: Heater |  |
|  | Clr: Cooler (0: Off, 1: On) |  |
|  | AP: Air Purifier (0: Off, 1: On) |  |
|  | V: Ventilation (0: Off, 1: On) |  |
|  | A: Alarm (0: Off, 1: On) |  |
|  |  |  |
|  |  |  |

## Inputs/Outputs

### A Verilog description example

module EnvironmentalControlSystem(

input wire humidity, // 1-bit Humidity Sensor Input (0: Optimal, 1: High)

input wire temperature, // 1-bit Temperature Sensor Input (0: Optimal, 1: Out of Range)

input wire air\_quality, // 1-bit Air Quality Sensor Input (0: Low, 1: High)

input wire dust\_allergen, // 1-bit Dust/Allergen Sensor Input (0: Low, 1: High)

output wire alarm, // 1-bit Alarm Output

output wire ventilation, // 1-bit Ventilation Output

output wire air\_purifier, // 1-bit Air Purifier Output

output wire dehumidifier // 1-bit Dehumidifier Output

);

// Combinational Logic for Outputs

assign alarm = (humidity || temperature || air\_quality || dust\_allergen) ? 1'b1 : 1'b0;

assign ventilation = (air\_quality) ? 1'b1 : 1'b0;

assign air\_purifier = (dust\_allergen) ? 1'b1 : 1'b0;

assign dehumidifier = (humidity) ? 1'b1 : 1'b0

Great! If we assume all inputs are binary variables, the combinational logic design becomes simpler. Each input will be either 0 or 1. Here is how you can define the environmental control system using Verilog with binary inputs:

### Verilog Description example 2

module EnvironmentalControlSystem(

input wire humidity, // 1-bit Humidity Sensor Input (0: Optimal, 1: High)

input wire temperature, // 1-bit Temperature Sensor Input (0: Optimal, 1: Out of Range)

input wire air\_quality, // 1-bit Air Quality Sensor Input (0: Low, 1: High)

input wire dust\_allergen, // 1-bit Dust/Allergen Sensor Input (0: Low, 1: High)

output wire alarm, // 1-bit Alarm Output

output wire ventilation, // 1-bit Ventilation Output

output wire air\_purifier, // 1-bit Air Purifier Output

output wire dehumidifier // 1-bit Dehumidifier Output

);

// Combinational Logic for Outputs

assign alarm = (humidity || temperature || air\_quality || dust\_allergen) ? 1'b1 : 1'b0;

assign ventilation = (air\_quality) ? 1'b1 : 1'b0;

assign air\_purifier = (dust\_allergen) ? 1'b1 : 1'b0;

assign dehumidifier = (humidity) ? 1'b1 : 1'b0;

endmodule

### Another example

module HomeAutomationSystem ( input wire humidity, // 1-bit Humidity Sensor Input (0: Optimal, 1: High) input wire temperature, // 1-bit Temperature Sensor Input (0: Optimal, 1: Out of Range) input wire air\_quality, // 1-bit Air Quality Sensor Input (0: Low, 1: High) input wire dust\_allergen, // 1-bit Dust/Allergen Sensor Input (0: Low, 1: High) output wire alarm, // 1-bit Alarm Output output wire ventilation, // 1-bit Ventilation Output output wire air\_purifier, // 1-bit Air Purifier Output output wire dehumidifier // 1-bit Dehumidifier Output ); // Combinational Logic for Outputs assign alarm = humidity || temperature || air\_quality || dust\_allergen; assign ventilation = air\_quality; assign air\_purifier = dust\_allergen; assign dehumidifier = humidity; endmodule

**Explanation:**

1. **Inputs:**
   * humidity: Binary input indicating the humidity level (0: Optimal, 1: High).
   * temperature: Binary input indicating the temperature level (0: Optimal, 1: Out of Range).
   * air\_quality: Binary input indicating the air quality (0: Low, 1: High).
   * dust\_allergen: Binary input indicating the dust/allergen level (0: Low, 1: High).
2. **Outputs:**
   * alarm: Activated if any sensor detects an undesirable condition.
   * ventilation: Activated if air quality is high.
   * air\_purifier: Activated if dust/allergen level is high.
   * dehumidifier: Activated if humidity level is high.

**Combinational Logic:**

* **Alarm:** Activated (1) if any of the sensor inputs are high (undesirable condition).
* **Ventilation:** Activated (1) if the air quality is high.
* **Air Purifier:** Activated (1) if the dust/allergen level is high.
* **Dehumidifier:** Activated (1) if the humidity level is high.