A Home Automation System Design Using Hardware Descriptive Tools

Kandepi Madhuri

B Likhith Sai

Bysani Sai Sirisha

Department of electronics and communication engineering, K L University

Abstract

Home automation involves introducing a degree of computerized or automatic control to certain electrical and electronic systems in a building. These include lighting, temperature control, security systems, controlling the door, etc. Today, safety and security is just a click of the appropriate technology away, and with such advancements happening, the security of one's home must also not be left behind. This project is an example of the use of digital systems design to accomplish that goal, where the security and comfort of a home is the issue at hand. This device had been modelled such that it takes care of home intrusion detection and avoidance, while it also controls other home environment factors such as temperature and smoke detection. A sequential pattern of controlling the door, burglar alarm, fire alarm, temperature and luminosity is followed in a priority order. The solution uses the hardware design system concepts of a state machine to design a Mealy system that is simulated in Verilog HDL using Xilinx and ModelSim. The expected output is achieved in the waveforms of the system, which is in agreement with the theoretical results expected and had realized the objective of controlling the home system.

Keywords: Verilog Hardware Descriptive Language, CPLD, Xilinx, ModelSim, state diagram

I. Introduction

1. Background

This applied project is a 'Home Automation System' using the design concepts for a hardware system. Due

to the number of thefts and increased risk of robbery, it is necessary to have a security system for home. 'Security' is the prime word in any business and since that being so important, is a very helpful and good topic for the benefit of mankind to develop a secure and safe system, which will prevent any break-ins to the home and also provide the comfort of controlling the temperature.

Such systems are already available in the market, suiting varying needs. But, if the consumer wants just one system, which can control the entire house, then there is a problem as there is no such equipment in the market yet. Home automation systems can control many things in a home, which are precious and valuable to a person. For example, the doors, windows, fire alarm and temperature can be controlled by just one click, just to ensure that all these are safe, when the occupants of the house are either at home or out or to just sit in one place and control these devices instead of having to get to that place to perform a certain action. The controller does its work of checking each of the devices periodically and in a certain order. The system can be controlled by a remote or any device that can be connected to all the devices that need to be secure. Their communication is through sensors which are attached on all the devices. This is a relatively new concept and has not hit the market so extensively yet, but once it does then it will definitely be very much in demand and so there will be a constant need for something so secure and safe.

2. Problem

This project is to design a controller, which provides an automated home security system at an affordable price. The parameter sensors connected to the controller will provide the required signals that will activate controller processes to and take the specified action.

3. Scope

A system is designed to control the doors, fire alarm, luminosity and the temperature. It is not intended or programmed to control anything else. This project is being implemented to control only these devices but can be expanded later on to control more devices or processes and also can be web enabled.

4. Existing Systems

Most commercially available home automation systems are all-in-one solutions which require that all controllable appliances are from the same company, or must be approved as compatible with said company's system. Moreover these systems normally come with a proprietary, dedicated device which acts as the control centre. To control the system from multiple locations, additional control devices must be purchased.

These complex systems usually need to be integrated when the building is constructed and must be planned in advance. They are also difficult to upgrade or replace once installed. The overall investment adds up considerably and is financially infeasible in most cases. These drawbacks hinder the popularity of such systems.

5. Proposed Systems

The objective of the proposed system is to offer a lowcost solution for a home automation system that overcomes the above drawbacks. The system provides basic control of appliances at a fraction of the cost of commercially available systems.

The concept of a proprietary control device is done away with as the system can be controlled from a Bluetooth or Wi-Fi enabled mobile device, such as a mobile phone or laptop. There is no need for a specialized server system as a typical desktop PC can act as the server. Nowadays most users already own the requisites such as a mobile phone and a desktop PC; hence the cost of the system is considerably reduced.

6. Modules

There are sensors for all four of the processes being automated. The sensors used are magnetic sensor (for doors), smoke detector (for fire), optical sensitive devices (for luminosity) and a thermostat (for temperature).

Every device will have a sensor that indicates how secure a system is by sending out a signal in case there is a change in the state.

The doors and windows operate on a magnetic sensor, if the signal on either is high (1) then the operation is to close the door. If the signal is a low (0), then the controller goes forward to check for the signal on the next device. There are separate magnetic sensors on both the front door and the rear door. The sensor on the fire alarm is a smoke detector, if it detects smoke then the signal goes to a high (1) and it triggers off the buzzer, else it goes to the next step.

The temperature controller has a thermostat on it that responds to with the required action depending on the temperature and then goes back to the beginning. The no-action temperature range is from $50^{\circ}F - 70^{\circ}F$. If the temperature goes below the range then the heater turns itself on, if it goes above the range then the air conditioner turns itself on. Similar is the case with the luminosity controller.

Once the program is dumped onto the CPLD Board, we interface the CPDL board with the sensor and buzzer circuits. After interfacing with external circuits, we test the overall performance of the system. The home automation system consists of several modules. Below is a brief description of each module.

Visitor Counter

For visitor counter, as soon as a person enter the door, the sensor detects the obstacle and feeds the entry event to the CPDL board which in turn drives the seven segment display to display the count of person entering the room according to the program dumped.

Burglar Alarm

For adding security to the home, we included a burglar alarm module in the system which buzzes when an unauthorized person enters the room.

Fire Alarm

The system comes along with a fire alarm module which buzzes when the room temperature crosses 105 degrees Celsius.

Luminosity Control

The system can control the ambient light in the room by turning on and off different light sources as and when needed.

Temperature Control

The system can automatically adjust the temperature inside the room by turning on various appliances when certain pre-configured temperatures are reached.

7. Approach

Following the hardware design concepts, the first step taken in designing this system is to figure out what equipment needs to be secure. This is the initial stage wherein the devices that need to be connected for security are short-listed and then they are put in a network and so the corresponding work is done. The devices are doors (doors include front and rear doors), windows, fire alarm and the temperature controller. Draw the bare bones block diagram that relates to all the devices, the sensors used and the inputs and the outputs. The block diagram showing all the inputs and outputs is drawn thus making it clear what inputs are required to drive which outputs.

As explained earlier, the signals on the attached sensors are captured and hence the required action is taken either on the same device or proceeds to the next device.

The Block diagram of the controller is as shown below where, the input signals are,

clk = clocks the device,

reset = resets,

 $d_{sen} = signal$ from the sensor on the Front Door,

 $f_{sen} = signal$ from the sensor for the fire alarm,

lum_sen = signal from the sensor on the luminosity controller.

temp_sen = signal from the sensor on the Temperature Controller.

and the output signals are,

b_alrm = HI -> close the front door, LO -> reset and wait for next command,

f alrm = buzz if fire alarm is on, i.e. HI,

 $t_{heat} = if$ temperature is less than 50 F turn heater on,

t_cool = if temperature is more than 70 F turn cooler on,

 $l_high = if$ luminosity is less than 50 lux, then increase the current flow through the light source.

 $l_low =$ if luminosity is high than 50 lux, then decrease the current flow through the light source.



The next step after the block diagram is to obtain the flow diagram of the same showing the different states and conditions. Initially, a simple flowchart is drawn in which the conditions and the required action to be taken are mentioned and then the detailed flowchart follows, in which the states and the devices will be shown.

The state diagram step is the next to follow in which all the states are shown and also see if the states can be reduced. A state diagram is developed that shows all the states and their behaviours. The states are examined for redundancy and reduced if any are found. This analysis and reduction helps to obtain a minimum and race free system. Simulate the system using Verilog HDL and see if it is working efficiently. The tools used for this purpose are the ModelSim and Xilinx. Based on the results obtained, the results are compared and verified with those that were assumed and hence the conclusions are obtained.

This report gives an in-depth knowledge of the applied project and its functionality along with the limitations, advantages, conclusions, future development and other factors. The report is divided into chapters and each chapter deals with one part of the applied project.

II. Theory

1) Verilog HDL

In the semiconductor and electronic design industry, Verilog is a hardware description language (HDL) used to model electronic systems. Verilog HDL, not to be confused with VHDL (a competing language), is most commonly used in the design, verification, and implementation of digital logic chips at the registertransfer level of abstraction. It is also used in the verification of analog and mixed-signal circuits.

Hardware description languages such as Verilog differ from software programming languages because they include ways of describing the propagation of time and signal dependencies (sensitivity). There are two assignment operators, a blocking assignment (=), and a non-blocking (<=) assignment. The non-blocking assignment allows designers to describe a statemachine update without needing to declare and use temporary storage variables. Since these concepts are part of Verilog's language semantics, designers could quickly write descriptions of large circuits in a relatively compact and concise form. At the time of Verilog's introduction (1984), Verilog represented a tremendous productivity improvement for circuit designers who were already using graphical schematic capture software and specially written software programs to document and simulate electronic circuits.

A Verilog design consists of a hierarchy of modules. Modules encapsulate design hierarchy, and communicate with other modules through a set of declared input, output, and bidirectional ports. Internally, a module can contain any combination of the following: net/variable declarations (wire, reg, integer, etc.), concurrent and sequential statement blocks, and instances of other modules (sub-hierarchies). Sequential statements are placed inside a begin/end block and executed in sequential order within the block. However, the blocks themselves are executed concurrently, making Verilog a dataflow language.

During the development cycle the description has to become more and more precise until it is actually possible to manufacture the product. The (automatic) transformation of a less detailed description into a more elaborated one is called synthesis. Existing synthesis tools are capable of mapping specific constructs of hardware description languages directly to the standard components of integrated circuits. This way, a formal model of the hardware system can be used from the early design studies to the final net list. Software support is available for the necessary refinement steps.

2) Flow Diagram

The first state is the start state, in this state all the inputs and outputs are set to 0.

Front door state: In this state the status of the front door is checked for and if the input from the magnetic sensor on the front door is 0 (LO) that indicates the door is closed, while if it is 1 (HI) then the door is open. If, II = 0 then the system goes to the next state else if II = 1 then the door is closed and the system goes to the next state which is the rear door state

Fire alarm state: The sensor here is the smoke detector and if it detects smoke then the input it sends is a 1 else it remains at 0. I4 = 0, the system goes to the next state meaning there is no fire, I4 = 1, means there is smoke and hence it triggers of a buzzer and goes to the next state which is the temperature state. The buzzer is on till it is manually turned off.

Temperature controller state: This state has 3 conditions to fulfil based on the input it receives. The thermostat on the temperature regulator keeps checking the temperature of the room. If it goes below 50° F, then the system turns on the heater and if temperature goes beyond 70° F, then the air conditioner is turned on. Else the system goes back to the start state. The specifications for this state are as follows: I5 > 70° F, then turn on the air conditioner else if I5 < 50° F, then

the heater is turned on. If the temperature is equal to 50° F or 70° F or anywhere between that span then no action is taken and the system goes back to the start state.

Luminosity control state: This state has 2 conditions to fulfil based on the input it receives. The Light Dependent Resistor (LDR) on the luminosity regulator keeps checking the light intensity of the room. If it goes below 1111, then the system increases the current through the light source and if it goes above 1111, then the system decreases the current through the light source. Else the system goes back to the start state. The specifications for this state are as follows: lum_sen > 00001111, then decrease the current through the light source else if lum_sen < 00001111, then the increase the current through the light source else if lum_sen < 00001111, then the increase the current through the light source are as follows: light source else the system goes back to the start state. The specifications for this state are as follows: lum_sen > 00001111, then decrease the current through the light source else if lum_sen < 00001111, then the increase the current through the light source. If the light intensity is equal to 00001111 then no action is taken and the system goes back to the start state.

3) State Diagrams



Current and next states

cur_st	nxt-st
Start	fdoor
Fdoor	fire

Fire	t_cool
t_cool	t_heat
t_heat	l_dim
I_dim	l_bright
l_bright	start

III. Implementation

Code

`timescale 1ns / 1ps // College: K L University // Engineer: Likhith, Madhuri, sirisha // Create Date: 20:26:00 03/25/2012 // Module Name: Home Automation // Project Name: Home Automation define start 4'd0 define fdoor 4'd2 define fire 4'd3 define t heat 4'd4 define t_cool 4'd5 define l_bright 4'd6 `define 1_dim 4'd7 `define comb lock 4'd8 module home_automation(clk,reset,din,d_sen,f_sen,temp_s en,lum_sen,

seg_out,b_alrm,f_alrm,heater,cooler,l_high,l_low); input clk,reset,d_sen,f_sen,din; input [7:0] temp_sen,lum_sen;

reg

output b_alrm,f_alrm,heater,cooler,l_high,l_low; output [7:0] seg_out;

reg [3:0] current_st; reg [3:0] next_st;

wire dclk; wire [3:0] bcd_data; debounce_filter m1 (.clk(clk), .rst (reset), .din(din), .dout(dclk)); bcd_counter m2 (.clk(dclk), .rst(reset), .q (bcd_data)); bcd_7seg m3 (.bcd_in(bcd_data), .sseg(seg_out));

initial begin current_st=`start; next_st=`start; b_alrm='b0; f_alrm='b0; heater='b0; cooler='b0; 1_high='b0; 1_low='b0; end always @(posedge clk) current_st=next_st; always @(current_st) begin case(current_st) `start: begin b_alrm='b0; f_alrm='b0; heater='b0; cooler='b0; 1_high='b0; l_low='b0; end `fdoor: b alrm='b1; `fire: f_alrm='b1; `t_heat: heater='b1; `t_cool: cooler='b1; `l_bright: l_low='b1; `l_dim: l_high='b1; endcase end always @(current_st,d_sen,f_sen,temp_sen,lum_sen,reset,d in) begin if(reset=='b1) next_st=`start; else case(current_st) `start: begin if(d_sen=='b1) next_st=`fdoor; else if(f_sen=='b1) next_st=`fire; else if(temp_sen > 'b1000110) next_st=`t_cool; else if(temp_sen < 'b1000110) next_st=`t_heat; else if(lum_sen > 'b1111) next_st=`l_dim; else if(lum_sen < 'b1111) next_st=`l_bright; end `fdoor: begin if(f_sen=='b1)

next_st=`fire; else if(temp_sen > 'b1000110) next_st=`t_cool; else if(temp_sen < 'b1000110) next_st=`t_heat; else if(lum_sen > 'b1111) next_st=`l_dim; else if(lum_sen < 'b1111) next_st=`l_bright; end `fire: begin if(temp_sen > 'b1000110) next_st=`t_cool; else if(temp_sen < 'b1000110) next_st=`t_heat; else if(lum_sen > 'b1111) next_st=`l_dim; else if(lum_sen < 'b1111) next_st=`l_bright; end `t_cool: begin if(temp_sen < 'b1000110) next_st=`t_heat; else if(lum_sen > 'b1111) next_st=`1_dim; else if(lum_sen < 'b1111) next_st=`l_bright; end `t_heat: begin if(lum_sen > 'b1111) next_st=`1_dim; else if(lum_sen < 'b1111) next_st=`l_bright; end `l_dim: begin if(lum_sen < 'b1111) next_st=`l_bright; end `l_bright: begin next_st=`start; end endcase end endmodule **Debounce filter** module debounce_filter(

input din,

```
input clk,
input rst,
output dout
);
    reg [2:0] temp;
    always @ (posedge clk or posedge rst)
    begin
    if (rst == 1'b1)
    temp <= 4'b000;
    else
    temp <= {temp[1:0], din};
    end
    assign dout = temp[0] & temp[1] & (!temp
[2]);
endmodule
```

BCD Counter

```
if (rst)

q \le 4'b0000;

else if (q == 4'b1001)

q \le 4'b0000;

else

q \le q + 1;

end

endmodule
```

BCD to 7 Segment Display

<pre>module bcd_7seg(input wire [3:0] bcd_in, output reg [7:0] sseg);</pre>		
always @ (bcd_in)		
begin		
case (bcd_in)		
4'b0000:	sseg	=
8'b10000001; //01111110;	-	
4'b0001:	sseg	=
8'b11001111; //00110000;	-	
4'b0010:	sseg	=
8'b10010010; //01101101;	-	

4'b0011:	sseg	=
8'b10000110; //01111001;		
4'b0100:	sseg	=
8'b11001100;//00110011;		
4'b0101:	sseg	=
8'b10100100; //01011011;		
4'b0110:	sseg	=
8'b10100000; //01011111;		
4'b0111:	sseg	=
8'b10001111; //01110000;		
4'b1000:	sseg	=
8'b10000000; //01111111;		
default : sseg	= 8'b100	00100;
//01111011;		
endcase		
end		

endmodule

IV. Results

1. Burglar Alarm

This is the simulation output when the burglar alarm goes off. That is, when an unauthorized person ties to enter through the door. This is achieved when d_sen is set to "1" (HIGH), this sets b_alrm to "1" (HIGH) as shown in the waveform below:



2. Fire Alarm

This is the simulation output when the fire alarm goes off. That is, when the thermistor is subjected to temperatures above 105° C. This is achieved when f_sen is set to "1" (HIGH), this sets f_alrm to "1" (HIGH) as shown in the waveform below:



3. Temperature Control

This is the simulation output of the temperature control module. The temperature control module controls the operation of both the cooler and the heater. When the temp_sen is greater than 1000110 then the cooler is turned on (cooler is set HIGH), as shown in the figure below:



and when temp_sen is lower than 1000110 then heater is turned on (heater is set HIGH) as shown in the figure below:

4. Luminosity Control

This is the simulation output of the luminosity control module. The luminosity control module controls the intensity of ambient light in the room. When the lum_sen is greater than 00001111 then the current through the light source is increased (l_high is set HIGH), as shown in the figure below:



and when lum_sen is lesser than 00001111 then the current through the light source is decreased (l_low is set HIGH), as shown in the figure below:



V. Conclusion

Concluding that the goal of this applied project has been reached, that is to design a controller, which provides an automated home security system at an affordable price. There have been many assumptions all through, but efforts have been put in to make it as practical as possible. The order of checking the devices based on the priority is followed and also the display shows the state the system is in. This is a low cost and effective device and with a few more changes it can be made a little more advance making it web enabled. This method is very easy to adapt and implement and can easily be embedded to another device.

I would also want to point out here that using Verilog HDL to implement the code part has really helped since it not only combines the hardware and software part, it also provides informative graphs and waveforms which are helpful in understanding the real concept of the project. Xilinx also has proved to be the most robust and learnable tool for simulation and a great integrated development environment.

Future Work

There are a few recommendations for making this system a marketable product.

Use the devices and the sensors and implement the hardware part. Convert this into a high level language and write it onto a chip creating a device that can be embedded to a PC, laptop, palm top etc. Make it web enabled to administer the system remotely. Introduce a few more devices to control, for example the cooking range, water temperature control, garage doors, and lights etc.

References

- Kim, Eung Soo, Kim, Min Sung, "Design and fabrication of security and home automation system", ICCSA 2006, International Conference Computational Science and Its Applications, Proceedings - Part III, 2006, pp. 31-37.
- [2] Fujiyama Hiroyuki, "System-on-a-chip with security modules for network home electric appliances" Fujitsu Scientific and Technical Journal, v 42, n 2 System-on-a-Chip, 2006, pp. 227-233.
- [3] A. Alheraish, "Design and Implementation of Home Automation System," IEEE Transactions on Consumer Electronics, vol. 50, no. 4, Nov. 2004, pp. 1087-1092.
- [4] Wobschall.D, Circuit Design For Electronic Instrumentation: Analog and Digital Devices from Sensor to Display, New York: Mcgraw-Hill, 1979.
- [5] VHDL Tutorial by Jan Van der Spiegel, University of Pennsylvania, Department of Electrical Engineering . Available: <u>http://www.seas.upenn.edu/~ee201/vhdl/vhdl prim</u> <u>er.html#_Toc526061341</u>
- [6] Thomas Petruzzellis, *The Alarm, Sensor & Security Circuit Cookbook*, Mcgraw-Hill, 1994
- [7] Ion Grout, Digital Systems Design with FPGAs and CPLDs, Elsevier Ltd, 2008