

SMART ROOM

Especially for kids!



12 Bahman 1399

Computer-Aided Digital System Design

Professor : Dr. Falahati

Mobina Kashanian - Amirhossein Ghasemi

Table Of Content

Smart room Especially for kids!



Table Of Content.....	2
• Introduction.....	4
• WHAT WE DESIGNED?	5
• ASM Description :	6
1.Password protection:	6
2.Check The Temperature:	7
3.Light Setup:.....	8
4.Window Alarm:	8
5.Baby is Crying!	9
6.Edge Of Bed:	9
7.Ventilation:	10
8.Moisture Of Bed:.....	10
• ASM Chart:.....	11
• FSM Chart:	12
• Verilog Implementation:	13
1.Check The Temperature:	13
2.Edge Of Bed	14
3.Baby is Crying!	15
4.Password protection	16
5.Open window.....	17

6.Light Setup.....	18
7.Moisture Of Bed	19
8.Ventilation	20
● ASIC Synthesis:.....	21
● FPGA Synthesis:.....	22



• **Introduction**

Kids are the most important part of a family that needs so much care.

In real life you have many works to do which makes you forget to do something that you have been planned for but now in this smart room we do something to take care of your room and house while you are busy enough to forget what you are about doing and also gives you an alert to do it yourself.

Kids in every age needs to be cared appropriate to their age. For example, when the baby feels hungry and wakes up, the mother must feed him.

In essence, the room should include things like room temperature and ventilation and keeping out of danger and keeping her health and the most important of her needs.

- **WHAT WE DESIGNED?**

We briefly outline the items that has been implemented in this project.

1. Parents can protect the child's room if necessary and specify a password for the door.
2. There are sensors in the room, and the first sensor is used to set the temperature,
and the parents can observe and determine the desired temperature.
3. A second sensor is used to set the proper light for the room during the day.
4. A third sensor is placed near the window and a certain sensor is lit to warn the parents if the child approaches the window and is in danger.(we have implemented an additional feature based on word detecting!)
5. Another sensor is embedded for the needs of the child and activates when the child is crying.
6. When a child is asleep, when it approaches the edge of the bed,
the sensor becomes active and automatically climbs onto the bed and prevents the child from falling.
7. Another sensor for ventilation, considered in two cases to enable it to be activated,
is the first mode in which the parents turn on, and the second mode is lit up
at a certain time.
8. We consider the sensor for the moisture of the bed.

- **ASM Description :**

We would prefer to describe the whole shape in general, in a complete detail at a glance.

1. Password protection:

First, the room is in its initial state with minimal facilities as we mentioned earlier the child's room has a password protection service that keeps the baby safe from any unpleasant events

The parents should enter the correct password to use other facility if the entered password

was not correct they can try again for two times, if their password matches and it was

correct so they can successfully enter to kid's room and they can use other facility but if

password was not correct and their attempted twice but the password does not match

the room will be set to the initial state with minimum facilities.

2. Check The Temperature:

In our designed room , we have a sensor to check the temperature of the child's room.

at first the sensor senses the environment temperature then based on it's value decides the air conditioner should be work on the cool or heat system.

As we mentioned above The air condition first senses the environment and then checks some condition and then decides if the air condition should work in cooler or heater mode.

The instruction works as follows ,if the environment temperature is between 18 to 24 centigrade the kids temperature is in a desired state.

but if the environment temperature is bigger than 24 centigrade, we also need to check an additional statement to decide the air conditioner speed, if the

difference between environment temperature and 24 centigrade is bigger than 10 centigrade then the air conditioner speed should be set to fast speed but if it's less than 10 centigrade then the speed should be set to slow.

If the environment temperature is less than 18 centigrade and the difference between environment and 18 is bigger than 10 the air conditioner should be in heater mode with fast speed , and it the difference is less than 10 the air conditioner is in heater mode with slow speed.

3. Light Setup:

Our designed light system is very smart! You may asking why we say this , it ' s because the light will be set properly based on light sensor and a clock.

Which means in the child ' s room we have a sensor which detects the light and decides the light should be turned on or off but it ' s not enough for a smart room so we decided to add and additional state to check the time, so if the time is between 11 PM to 7 AM the light should be turned off and the curtain is closed and if the time is between 7 AM to 3 PM the lights should be off and the curtain should be open the reason of this action is the sun is up and we can save the energy and use sunlight.

If the sensor senses that the child ' s room needs light and the clock was in 3 PM to 11 PM the light should be turned on and the curtain should be closed.

4. Window Alarm:

If the baby came near the window, the window will make an alarm to warn parents.

This caution is based on the distance of the baby.

And also we have an additional smart feature for window and it ' s for opening and closing windows based on a special word detection , if the parents says the word “BAZ” the window will be open automatically.

5. Baby is Crying!

When the baby is crying we will make an alarm to warn parents.

The sensor will be work when it's detect a special word in our design for this small room we set the special word "OWA", because most of the babies while crying said that word so when the sensor detect this word it will start an alarm to warn to parents.

6. Edge Of Bed:

When the baby is sleeping and came to edge of bed we will make an alarm to warn parents and prevent baby's falling.

The sensor is set on the edge of bed so when the baby's body hit that sensor and sensor senses the baby's body it will start to make an alarm to parents.

7. Ventilation:

The used a sensor for check the air quality, which has to be controlled every 30 minutes.

we used a counter, which is activated when it's meet the defined deadline which is set to be 30 minute.

If the ventilation was needed, the ventilation system will be turned on and if it's not necessary it will be switched off.

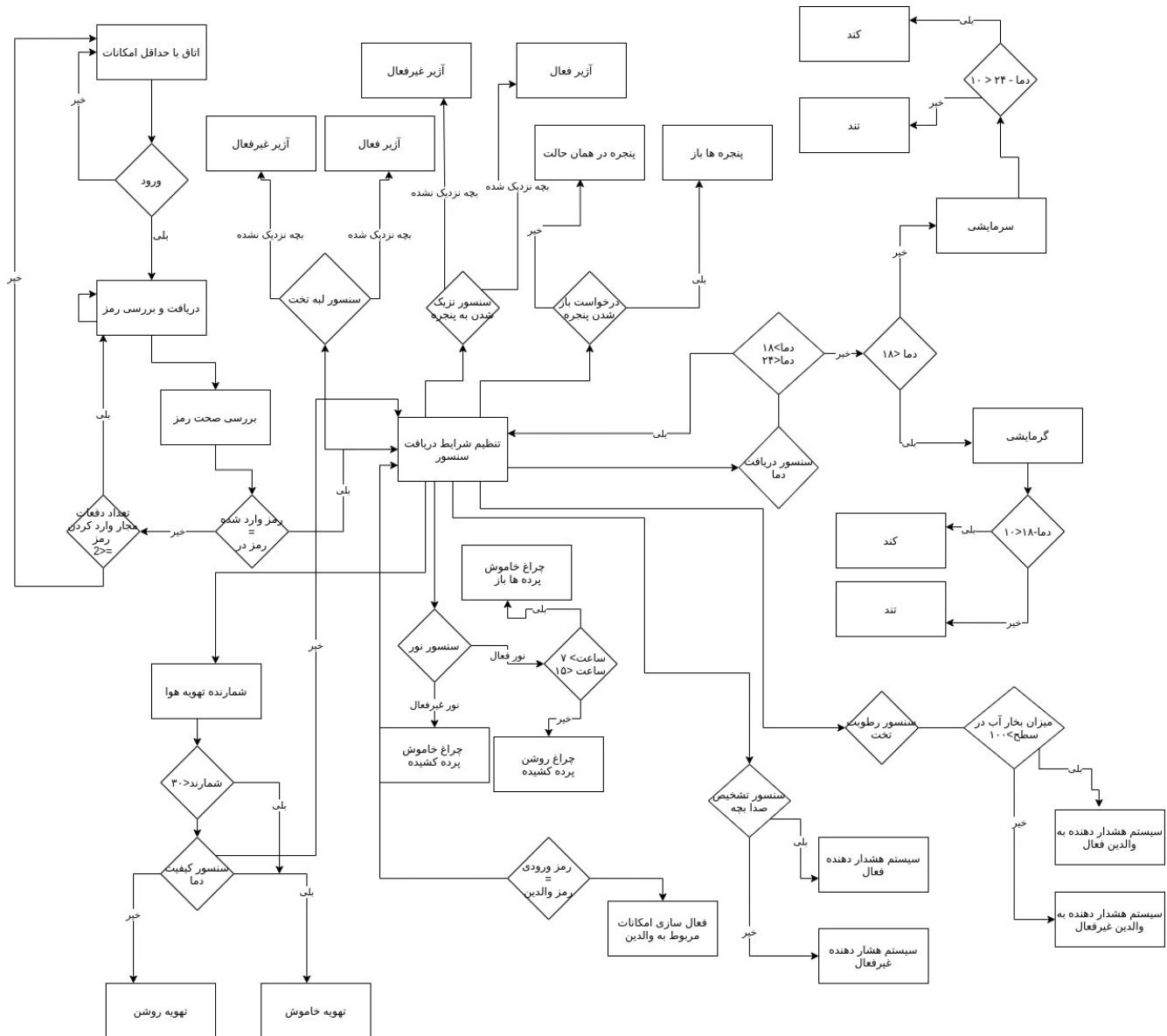
8. Moisture Of Bed:

When the baby wet their beds a sensor senses the water vapor rate and if this rate is less than 100 the parents can find out that the baby's need their parents to change their diaper or help them to clean their bed .

- **ASM Chart:**

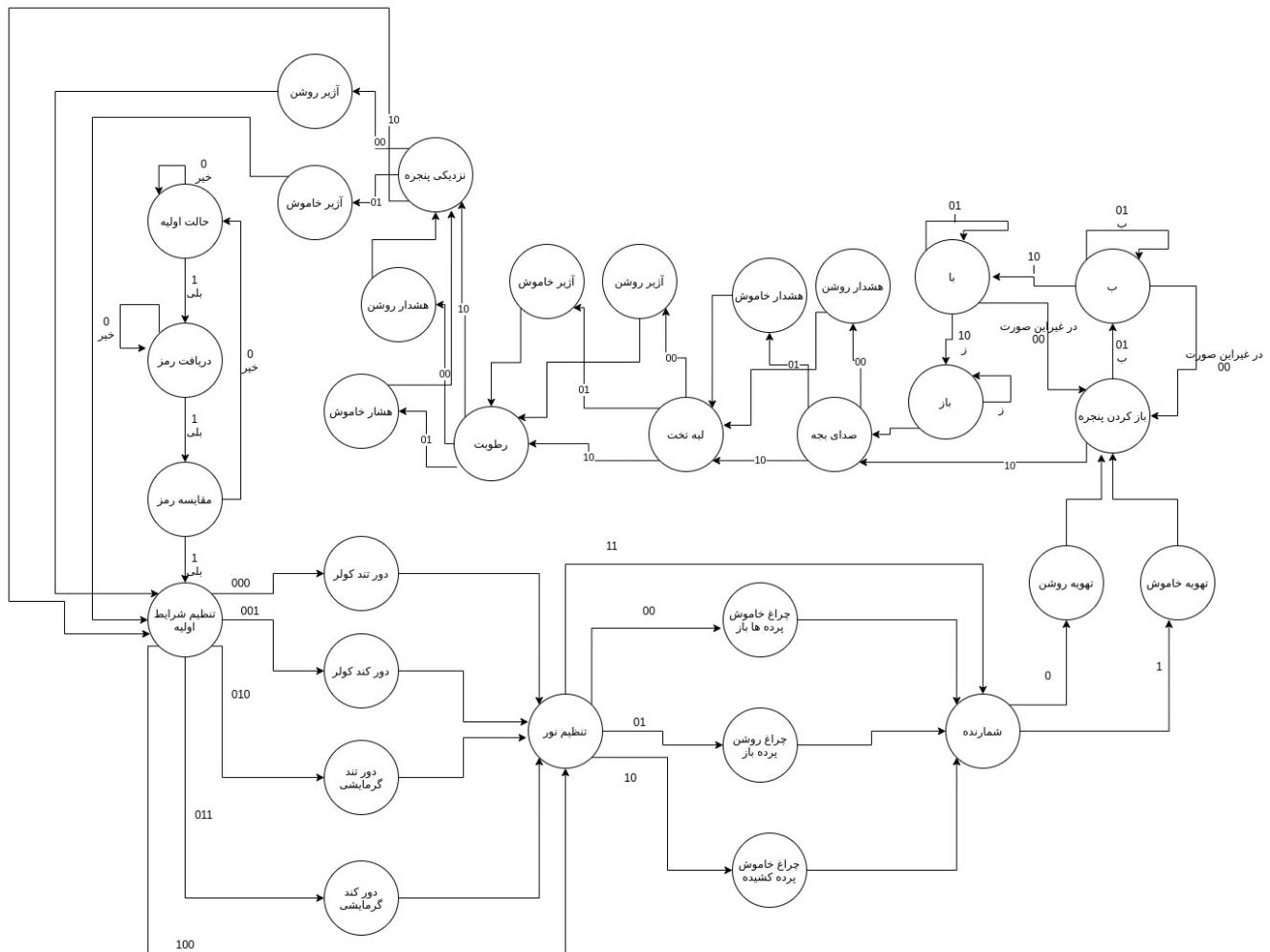
The ASM chart attached as follows.

(if the quality of the image is not so well you can find the image in the document folder of project)



• FSM Chart:

Well , as you know ASM and FSM very similar to each other so we ' ll not describe this chart in much detail , we just cover important aspects.



• Verilog Implementation:

For each items that described in features list we have Verilog implementation.

We have also implemented the test benches and Wave form.

1. Check The Temperature:

The screenshot displays two windows of the ISE Project Navigator interface, showing Verilog code for a temperature control system.

Left Window (AirConditioning.v):

```
1 module AirConditioning(temp, ideal, clk, out_temp);
2   input [5:0] ideal;
3   input clk;
4
5   output reg [5:0] out_temp;
6
7   always @(posedge clk)
8     begin
9       if (temp < ideal)
10         out_temp = temp + 1;
11       else if (temp > ideal)
12         out_temp = temp - 1;
13       else
14         out_temp = temp;
15     end
16
17 endmodule
18
19
20
```

Right Window (AirConditioningTest.v):

```
1 timescale 1ns / 1ps
2
3 module AirConditioningTest;
4
5   // Inputs
6   reg [5:0] temp;
7   reg [5:0] ideal;
8   reg clk;
9
10  // Outputs
11  wire [5:0] out_temp;
12
13  // Instantiate the Unit Under Test (UUT)
14  AirConditioning uut (
15    .temp(temp),
16    .ideal(ideal),
17    .clk(clk),
18    .out_temp(out_temp)
19  );
20
21 initial begin
22   // Initialize Inputs
23   temp = 0;
24   ideal = 0;
25   clk = 0;
26
27   // Wait 100 ns for global reset to finish
28   #100;
29
30   // Add stimulus here
31   ideal = 6'd27;
32   temp = 6'd17; // temperature is lower than ideal temp
33
34   #10 clk = ~clk;
35   #3; temp = out_temp;
36
37 end
```

2. Edge Of Bed

ISE Project Navigator (P.20131013) - C:\Users\Mobina\Downloads\Telegram Desktop\Cad_project_96522321_97522184\Cad_project_96522321_97522184\verilog\cad_project\xise - [BedAlarm.v]

The screenshot shows the ISE Project Navigator interface with the BedAlarm.v file open. The code defines a module BedAlarm with an input sensor, an input clk, and an output reg out_alarm. It contains an always block that checks the sensor value and updates the out_alarm output accordingly. The ISE navigation pane on the left shows the project hierarchy and current processes.

```
1 timescale 1ns / 1ps
2 module BedAlarm(sensor, clk, out_alarm);
3   input sensor;
4   input clk;
5   output reg out_alarm;
6
7   always @(sensor)
8     begin
9       if (sensor == 1)
10         out_alarm = 1;
11       else
12         out_alarm = 0;
13     end
14
15 endmodule
16
```

ISE Project Navigator (P.20131013) - C:\Users\Mobina\Downloads\Telegram Desktop\Cad_project_96522321_97522184\Cad_project_96522321_97522184\verilog\cad_project\xise - [BabyAlarmTest.v]

The screenshot shows the ISE Project Navigator interface with the BabyAlarmTest.v file open. The code defines a module BabyAlarmTest with inputs sensor and clk, and an output out_alarm. It includes a comment indicating the instantiation of the Unit Under Test (UUT) as a BedAlarm module. The ISE navigation pane on the left shows the project hierarchy and current processes.

```
1 timescale 1ns / 1ps
2
3
4 module BabyAlarmTest;
5
6   // Inputs
7   reg sensor;
8   reg clk;
9
10  // Outputs
11  wire out_alarm;
12
13  // Instantiate the Unit Under Test (UUT)
14  BedAlarm uut (
15    .sensor(sensor),
16    .clk(clk),
17    .out_alarm(out_alarm)
18  );
19
20  initial begin
21    // Initialize Inputs
22    sensor = 0;
23    clk = 0;
24
25    // Wait 100 ns for global reset to finish
26    #100;
27
28    // Add stimulus here
29
30    #10 clk = ~clk;
31    #8 sensor =~sensor;
32    $display(sensor);
33    $display(out_alarm);
34    #10 clk = ~clk;
```

3. Baby is Crying!

ISE Project Navigator (P.20131013) - C:\Users\Mobina\Downloads\Telegram Desktop\Cad_project_96522321_97522184\Cad_project_96522321_97522184\verilog\cad_project\xise - [BabyCry.v]

```
1 `timescale 1ns / 1ps
2 module BabyCry( input clk, input rstn, input in, output out );
3
4 parameter IDLE = 0, SO = 1, SOW = 2, SOWA = 3;
5
6 reg [2:0] cur_state, next_state;
7
8 assign out = cur_state == SOWA ? 1 : 0;
9
10 always @ (posedge clk) begin
11   if (!rstn)
12     cur_state <= IDLE;
13   else
14     cur_state <= next_state;
15 end
16
17 always @ (cur_state or in) begin
18   case (cur_state)
19     IDLE : begin
20       if (in)
21         next_state = SO;
22       else
23         next_state = IDLE;
24     end
25     SO : begin
26       if (in)
27         next_state = IDLE;
28       else
29         next_state = SOW;
30     end
31     SOW : begin
32       if (in)
33         next_state = SOWA;
34       else
```

ISE Project Navigator (P.20131013) - C:\Users\Mobina\Downloads\Telegram Desktop\Cad_project_96522321_97522184\Cad_project_96522321_97522184\verilog\cad_project\xise - [BabyCryTest.v]

```
1 `timescale 1ns / 1ps
2
3 module BabyCryTest;
4
5 // Inputs
6 reg clk;
7 reg rstn;
8 reg[3:8:0] in;
9 always #10 clk =~clk;
10 // Outputs
11 wire out;
12
13 // Instantiate the Unit Under Test (UUT)
14 BabyCry uut (
15   .clk(clk),
16   .rstn(rstn),
17   .in(in),
18   .out(out)
19 );
20
21 initial begin
22   // Initialize Inputs
23   clk <= 0;
24   in <=1;
25   rstn <=0;
26   repeat (5) @(posedge clk);
27   rstn <=1;
28
29   // Add stimulus here
30
31   @(posedge clk) in<="B";
32   @(posedge clk) in<="A";
33   @(posedge clk) in<="N";
34   @(posedge clk) in<="K";
```

4. Password protection

ISE Project Navigator (P.20131013) - C:\Users\Mobina\Downloads\Telegram Desktop\Cad_project_96522321_97522184\Cad_project_96522321_97522184\verilog\cad_project\xise - [Door.v]

```
1 timescale 1ns / 1ps
2 module Door(ipass0, pass1, pass2, pass3, ipass0, ipass1, ipass2, ipass3, switch, unlock
3 );
4   input [3:0] ipass0;
5   input [3:0] pass1;
6   input [3:0] pass2;
7   input [3:0] pass3;
8
9   input [3:0] ipass0;
10  input [3:0] ipass1;
11  input [3:0] ipass2;
12  input [3:0] ipass3;
13
14  input switch;
15  output reg unlock;
16
17 always @(ipass0 or ipass1 or ipass2 or ipass3)
18 begin
19   if (switch == 1)
20     if (ipass0 == ipass0 & pass1 == ipass1 & pass2 == ipass2 && pass3 == ipass3)
21       begin
22         unlock = 1;
23         #100;
24         unlock = 0;
25       end
26     else
27       unlock = 0;
28     else
29       unlock = 1;
30   end
31
32 endmodule
33
```

ISE Project Navigator (P.20131013) - C:\Users\Mobina\Downloads\Telegram Desktop\Cad_project_96522321_97522184\Cad_project_96522321_97522184\verilog\cad_project\xise - [DoorTest.v]

```
1 timescale 1ns / 1ps
2 module DoorTest;
3
4   // Inputs
5   reg [3:0] pass0;
6   reg [3:0] pass1;
7   reg [3:0] pass2;
8   reg [3:0] pass3;
9   reg [3:0] ipass0;
10  reg [3:0] ipass1;
11  reg [3:0] ipass2;
12  reg [3:0] ipass3;
13
14  reg switch;
15
16  // Outputs
17  wire unlock;
18
19 // Instantiate the Unit Under Test (UUT)
20 Door uut (
21   .pass0(pass0),
22   .pass1(pass1),
23   .pass2(pass2),
24   .pass3(pass3),
25   .ipass0(ipass0),
26   .ipass1(ipass1),
27   .ipass2(ipass2),
28   .ipass3(ipass3),
29   .switch(switch),
30   .unlock(unlock)
31 );
32
33 initial begin
34   // Initialize Inputs
35
```

5. Open window

The screenshot shows the ISE Project Navigator interface. The left pane displays a hierarchical tree under 'Design' labeled 'Implementation'. It includes nodes for 'AirConditioningTest', 'BabyCryTest', 'DoorTest', 'det_baz_test', 'lightTest', 'moistureTest', and 'ventilationTest'. The 'det_baz_test' node is currently selected. The right pane contains a code editor with the following Verilog-like logic:

```
1 timescale 1ns / 1ps
2
3 module det_baz ( input clk, input rstn, input in, output out );
4
5 parameter IDLE = 0, Sb = 1, Sba = 2, Sbaz = 3;
6
7 //parameter IDLE = 0, SO = 1, SW= 2, SA = 3; if the baby cries the sound like "Owa"
8
9 reg [2:0] cur_state, next_state;
10
11 assign out = cur_state == Sbaz ? 1 : 0;
12
13 always @ (posedge clk) begin
14   if (!rstn)
15     cur_state <= IDLE;
16   else
17     cur_state <= next_state;
18 end
19
20 always @ (cur_state or in) begin
21   case (cur_state)
22     IDLE : begin
23       if (in)
24         next_state = Sb;
25       else
26         next_state = IDLE;
27     end
28     Sb : begin
29       if (in)
30         next_state = IDLE;
31       else
32         next_state = Sba;
33     end
34     Sba : begin
```

The screenshot shows the ISE Project Navigator interface with the following details:

- Project Path:** C:\Users\Mobina\Downloads\Telegram Desktop\Cad_project_96522321_97522184\Cad_project_96522321_97522184\verilog\cad_project\cad_project.xise - {det_baz_test.v}
- File Menu:** File, Edit, View, Project, Source, Process, Tools, Window, Layout, Help
- Design View:** Implementation, Simulation, Behavioral, Hierarchy.
- Hierarchy Tree:** AirConditioningTest (AirConditioningTest.v), BabyAlarmTest (BabyAlarmTest.v), BabyCryTest (BabyCryTest.v), DoorTest (DoorTest.v), det_baz_test (det_baz_test.v), lightTest (lightTest.v), moistureTest (moistureTest.v), ventilationTest (ventilationTest.v).
- Processes:** No Processes Running.
- Simulator:** ISim Simulator, Behavioral Check Syntax, Simulate Behavioral Model.
- Code Editor Content:** The code is a Verilog testbench for a module named `det_baz_test`. It includes declarations for timescale, inputs (clk, rstn), and outputs (out_res). It instantiates the UUT (det_baz) and provides stimulus by setting clk and rstn values over 5 clock cycles.

```
1 timescale 1ns / 1ps
2
3
4 module det_baz_test;
5
6 // Inputs
7 reg clk;
8 reg rstn;
9 reg[3:8:0] in;
10 always #10 clk =~ clk;
11
12 // Outputs
13 wire out_res;
14
15 // Instantiate the Unit Under Test (UUT)
16 det_baz uut (
17 .clk(clk),
18 .rstn(rstn),
19 .in(in),
20 .out(out_res)
21 );
22
23 initial begin
24 // Initialize Inputs
25 clk <= 0;
26 in <=1;
27 rstn <=0;
28 repeat (5) @(posedge clk);
29 rstn <=1;
30
31 // Add stimulus here
32
33 @(posedge clk) in<="BB";
34 @(posedge clk) in<="AA";
```

6. Light Setup

ISE Project Navigator (P.20131013) - C:\Users\Mobina\Downloads\Telegram Desktop\Cad_project_96522321_97522184\Cad_project_96522321_97522184\verilog\cad_project\xise - [LightSystem.v]

```
1 `timescale 1ns / 1ps
2
3 module LightSystem(
4     input clk,
5     input reset,
6     input wire StartOn,
7     input wire StartOff,
8     input wire Keypad,
9     input wire state,
10
11     output reg initialize ,
12     output reg lamp_on,
13     output reg lamp_off,
14     output reg start_on_turn_on_button,
15     output reg start_on_turn_off_button,
16     output reg timingpass,
17     output reg lamp_stays_off,
18     output reg lampstate
19 );
20
21 parameter STATE0='b000,
22     STATE1='b001,
23     STATE2='b010,
24     STATE3='b011,
25     STATE4='b100,
26     STATE5='b101,
27     STATE6='b110,
28     STATE7='b111;
29
30 reg[2:0] state_reg,next_state ;
31
32 always @(posedge clk, posedge reset)
33 begin
34
```

ISE Project Navigator (P.20131013) - C:\Users\Mobina\Downloads\Telegram Desktop\Cad_project_96522321_97522184\Cad_project_96522321_97522184\verilog\cad_project\xise - [lightTest.v]

```
1 `timescale 1ns / 1ps
2
3
4 module lightTest;
5
6 // Inputs
7 reg clk;
8 reg reset;
9 reg StartOn;
10 reg StartOff;
11 reg keypad;
12 reg state;
13
14 // Outputs
15 wire initialize;
16 wire lamp_on;
17 wire lamp_off;
18 wire start_on_turn_on_button;
19 wire start_on_turn_off_button;
20 wire timingpass;
21 wire lamp_stays_off;
22 wire lampstate;
23
24 // Instantiate the Unit Under Test (UUT)
25 LightSystem uut (
26     .clk(clk),
27     .reset(reset),
28     .StartOn(StartOn),
29     .StartOff(StartOff),
30     .Keypad(keypad),
31     .state(state),
32     .initialize(initialize),
33     .lamp_on(lamp_on),
34
```

7. Moisture Of Bed

The screenshot shows the ISE Project Navigator interface with the following details:

- Project Path:** C:\Users\Mobina\Downloads\Telegram Desktop\Cad_project_96522321_97522184\Cad_project_96522321_97522184\verilog\cad_project\cad_project.xise - [moisture.v]
- File Menu:** File, Edit, View, Project, Source, Process, Tools, Window, Layout, Help
- Design View:** Implementation, Simulation, Behavioral, Hierarchy.
- Hierarchy Tree:** BabyAlarmTest (BabyAlarmTest.v), BedAlarm (BedAlarm.v), BabyCryTest (BabyCryTest.v), BabyCry (BabyCry.v), DoorTest (DoorTest.v), Door (Door.v), det_baz_test (det_baz_test.v), lightTest (lightTest.v), lightSystem (LightSystem.v), moistureTest (moistureTest.v), moisture (moisture.v), ventilationTest (ventilationTest.v).
- Processes:** No Processes Running.
- Process List:** ISim Simulator, Behavioral Check Syntax, Simulate Behavioral Model.
- Code Editor:** The main pane displays the Verilog code for the 'moisture' module:

```
1 timescale 1ns / 1ps
2
3 module moisture(wet, ideal, clk,alarm);
4     input [5:0] ideal;
5     input clk;
6
7     input [5:0] wet;
8     output reg [5:0] alarm;
9
10
11    always @(posedge clk)
12        begin
13            if (wet <= ideal)
14                alarm = 0;
15            else
16                alarm = 1;
17        end
18
19
20 endmodule
21
22
```

The screenshot shows the ISE Project Navigator interface with the following details:

- File Menu:** File, Edit, View, Project, Source, Process, Tools, Window, Layout, Help.
- Design View:** Implementation, Simulation.
- Hierarchy:** BabyAlarmTest (BabyAlarmTest.v), BabyCryTest (BabyCryTest.v), DoorTest (DoorTest.v), det_baz_test (det_baz_test.v), lightTest (lightTest.v), moistureTest (moistureTest.v), ventilationTest (ventilationTest.v).
- Processes:** moistureTest
- Selected Item:** iSim Simulator - Behavioral Check Syntax.
- Code Editor:** The code is a Verilog testbench for a moisture sensor. It includes declarations for湿 (wet), ideal湿 (ideal), and clock (clk) inputs; an alarm output; and a UUT instantiation. It also includes an initial block for setting initial values and a comment for adding stimulus.

```
1 timescale 1ns / 1ps
2
3
4 module moistureTest;
5
6   // Inputs
7   reg [5:0] wet;
8   reg [5:0] ideal;
9   reg clk;
10
11   // Outputs
12   wire [5:0] alarm;
13
14   // Instantiate the Unit Under Test (UUT)
15   moisture uut (
16     .wet(wet),
17     .ideal(ideal),
18     .clk(clk),
19     .alarm(alarm)
20 );
21
22 initial begin
23   // Initialize Inputs
24   wet = 0;
25   ideal = 0;
26   clk = 0;
27
28   // Add stimulus here
29
30
31   #200
32   clk = ~clk;
33   #100;
34   ideal = 6'd27;
```

8. Ventilation

ISE Project Navigator (P20131013) - C:\Users\Mobina\Downloads\Telegram Desktop\Cad_project_96522321_97522184\Cad_project_96522321_97522184\verilog\cad_project\cad_project.xise - [ventilation.v]

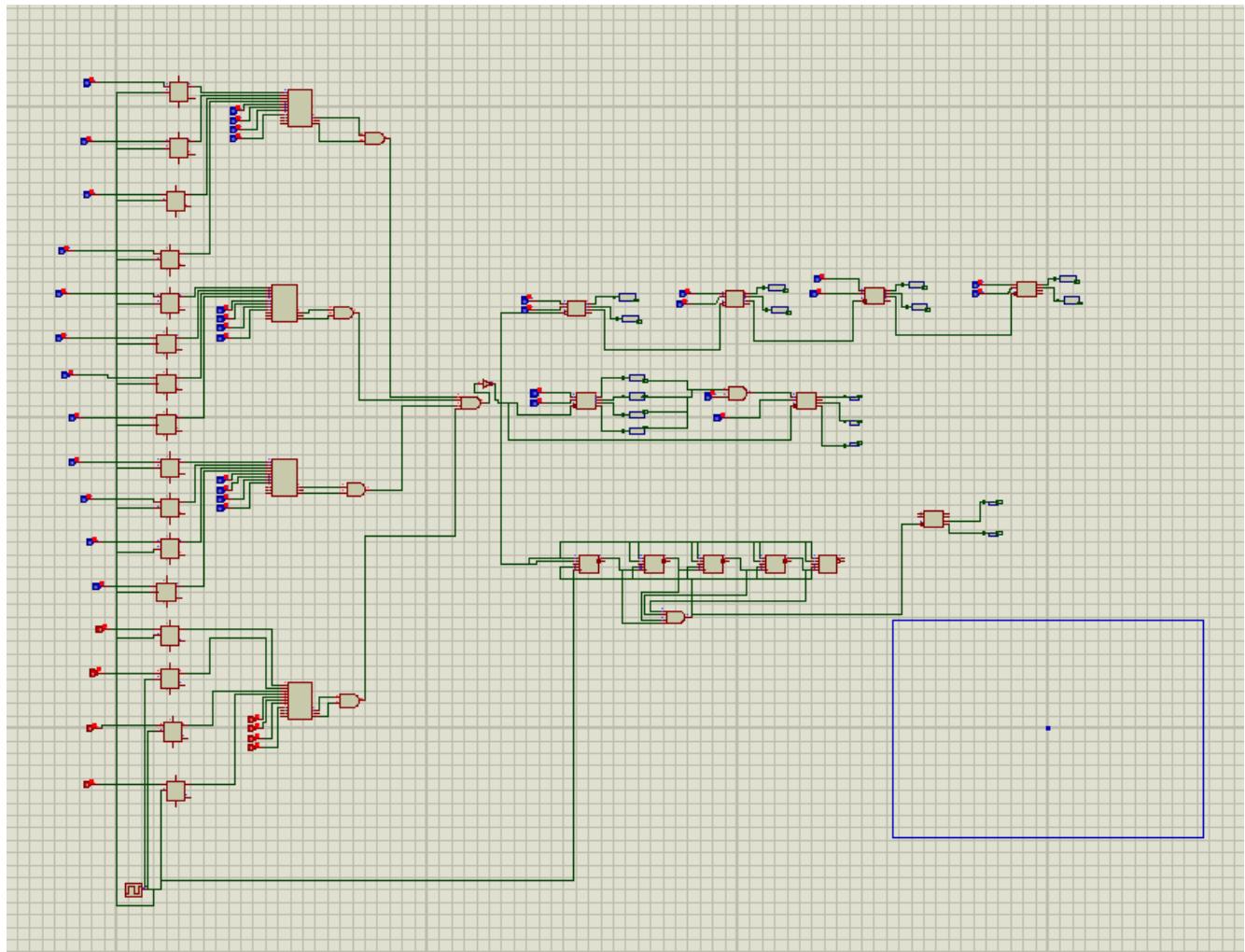
```
1 `timescale 1ns / 1ps
2 module ventilation( clk , rst , vintlation_sensor , counter , state_vintilation);
3
4   input clk;
5   input vintlation_sensor;
6
7   output reg rst;
8   output reg[3:0] counter;
9   output reg state_vintilation;
10
11
12   always @(posedge clk) begin
13     if(rst == 1) begin
14       counter <= 3'b000;
15     end
16     if(vintlation_sensor==1)
17       state_vintilation = 1;
18     else
19       state_vintilation = 0;
20   end
21
22   if(counter==4'd30) begin
23     rst = 1;
24   end
25   counter <= counter+1;
26 end
27 endmodule
28
29
30
31
32
33
34
```

ISE Project Navigator (P20131013) - C:\Users\Mobina\Downloads\Telegram Desktop\Cad_project_96522321_97522184\Cad_project_96522321_97522184\verilog\cad_project\cad_project.xise - [ventilationTest.v]

```
1 `timescale 1ns / 1ps
2 module ventilationTest;
3
4   // Inputs
5   reg clk;
6   reg vintlation_sensor;
7
8   // Outputs
9   wire rst;
10  wire [3:0] counter;
11  wire state_vintilation;
12
13  // Instantiate the Unit Under Test (UUT)
14  ventilation uut (
15    .clk(clk),
16    .rst(rst),
17    .vintlation_sensor(vintlation_sensor),
18    .counter(counter),
19    .state_vintilation(state_vintilation)
20  );
21
22  initial begin
23    // Initialize Inputs
24    clk = 0;
25    vintlation_sensor = 0;
26    // rst <1;
27    // counter <= 3'b000;
28
29    // Add stimulus here
30
31
32
33
34    #10
35
```

- **ASIC Synthesis:**

We used ASIC implementation. We used proteus to design all the component, and it's working exactly as we expected .



- **FPGA Synthesis:**

We used FPGA implementation.

