

## **Project Topic : Room Automation**

• Subject : **FPGA (Field Programmable Gate Array)**

• Department : **BE(CSE)**

• Semester: **5nd-Semester(FA20)**

• Submitted By : **Adeena Shahnawaz. FA20-BECE-0019**

• Submitted To: **Dr. Areeb Ahmed**

**CONTENTS:**

* **Object.**
* **Methodology.**
* **Coding.**
* **Test bench.**
* **Simulation and Circuit Diagram.**
* **Conclusion.**

**OBJECT :**

In this project, the classroom has been mechanized by using various gates, including; and, or, xor. We have a number of parts, such as Light (8), Fans (4), AC (4), and Projector (1), and requirements are given for each part (input). If any requirement is not fulfilled, the bulb will be switched on impulsively.

Following Basic Requirements are needed for this project:

1.There won't be a problem if there are two air conditioners on, but if there are more than two, the bulb will be brightened unnaturally .

2.If three fans are in use, there won't be a problem, but if the working fans are more than three then the bulb will automatically be on .

3.If less than three air conditioners are being used then everything will be ok.On contrary, working with three or more than three air conditioners will compel the bulb to be on.

4.Working with 5 LIGHTS will be perfect. Incase of increasing their numbers will enlighten the bulb automatically.

5.Incase of switched off every appliance with switched on projector or switched on every appliance with switched off projector,bulb will be brighten.

**METHODOLOGY:**

First, we use a k-map to create equations for our desired output. Later, we combine and, or, gates to create a (AC) circuit. We then use and/or gates to create a (FAN) circuit, and we combine gates to create a (LIGHT) circuit. Additionally, we include a (PROJECT) circuit, generate a possible answer by combining a number of gates, and then merge the entire project with an or gate.

**CODING:**

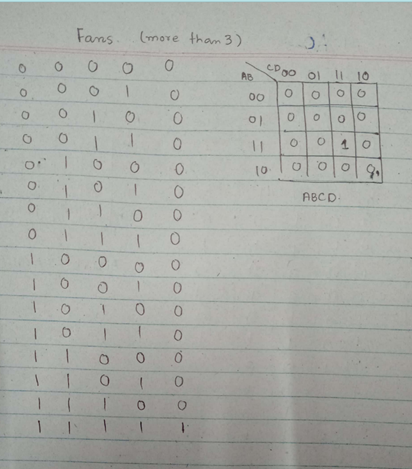
| library ieee;  use ieee.std\_logic\_1164.all;  entity room\_automation is  port(  l : in std\_logic\_vector(16 downto 0); ---for light (7 downto 0) , for fan (11 downto 8) , for ac (15 downto 12) , for projector (16)  result : out std\_logic  );  end room\_automation;  architecture sop\_arch of room\_automation is  signal s1 , s2 , s3 , s4 , s5 , s6 , s7 , s8 , s9 , s10 , s11 , s12 , s13 , s14 , s15 , s16 , s17 , s18, acout, fanout, project1, lightout ,sout : std\_logic;  begin  s1 <=l(0) and l(1) and l(2) and l(3) and l(4) and l(5);  s2 <=l(0) and l(4) and l(5) and l(6) and l(7);  s3 <=l(0) and l(2) and l(4) and l(5) and l(7);  s4 <=l(1) and l(3) and l(4) and l(5) and l(7);  s5 <=l(0) and l(1) and l(2) and l(3) and l(7);  s6 <=l(0) and l(2) and l(4) and l(5) and l(6);  s7 <=l(0) and l(1) and l(3) and l(4) and l(6);  s8 <=l(1) and l(2) and l(3) and l(5) and l(6);  s9 <=l(0) and l(1) and l(4) and l(6) and l(7);  s10 <=l(1) and l(2) and l(5) and l(6) and l(7);  s11 <=l(1) and l(3) and l(4) and l(6) and l(7);  s12 <=l(0) and l(1) and l(3) and (not l(4)) and l(5) and l(6) and l(7);  s13 <=(not l(0)) and (not l(1)) and l(2) and l(3) and l(4) and l(5) and l(6) and l(7);  s14 <=(not l(0)) and l(1) and l(2) and l(3) and l(4) and (not l(5)) and l(6) and l(7);  s15 <=(not l(0)) and l(1) and l(2) and l(3) and (not l(4)) and l(5) and l(6) and l(7);  s16 <=l(0) and (not l(1)) and l(2) and l(3) and (not l(4)) and (not l(5)) and l(6) and l(7);  s17 <=l(0) and (not l(1)) and l(2) and l(3) and l(4) and (not l(5)) and l(6) and l(7);  s18 <=(not l(0)) and (not l(1)) and l(2) and l(3) and (not l(4)) and l(5) and l(6) and l(7);  lightout <=s1 or s2 or s3 or s4 or s5 or s6 or s7 or s8 or s9 or s10 or s11 or s12 or s13 or s14 or s15 or s16 or s17 or s18;  acout <= (l(12) and l(13) and l(15)) or (l(12) and l(13) and l(14)) or (l(12) and l(14) and l(15)) or (l(13) and l(14) and l(15));  fanout <= l(8) and l(9) and l(10) and l(11);  sout <= l(0) or l(1) or l(2) or l(3) or l(4) or l(5) or l(6) or l(7) or l(8) or l(9) or l(10) or l(11) or l(12) or l(13) or l(14) or l(15);  project1 <= sout xor (not l(16));  result <= acout or fanout or lightout or project1 ;  end sop\_arch; |
| --- |

**TEST BENCH :**

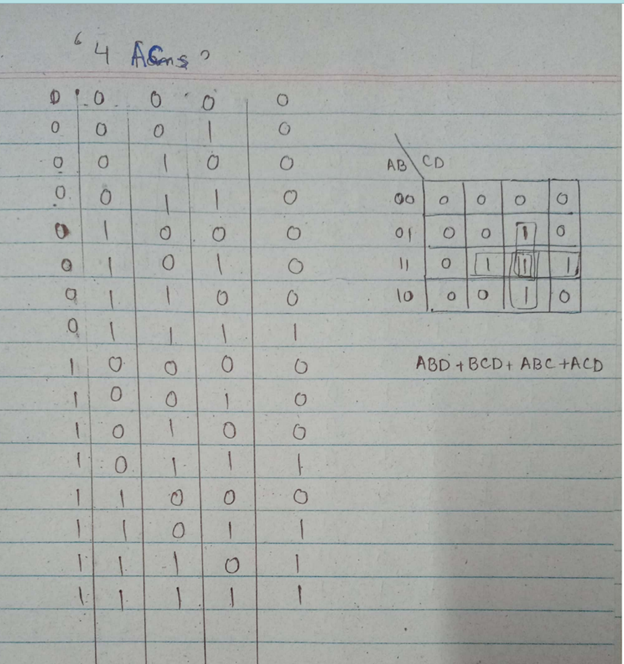
| library ieee;  use ieee.std\_logic\_1164.all;  entity room\_automation\_testbench is  end room\_automation\_testbench;  architecture tb\_arch of room\_automation\_testbench is  component room\_automate  port(  a : in std\_logic\_vector(16 downto 0);  result : out std\_logic  );  end component;  signal test\_in : std\_logic\_vector(16 downto 0);  signal test\_out : std\_logic;  begin  uut : room\_automation  port map(all\_input => test\_in , result => test\_out);  process  begin  test\_in <= "00000000000000000";  test\_in <= "00000000000000001";  test\_in <= "00000000000000010";  test\_in <= "00000000000000011";  test\_in <= "00000000000000100";  test\_in <= "00000000000000101";  ---.......  test\_in <= "11111111111111100";  test\_in <= "11111111111111101";  test\_in <= "11111111111111110";  test\_in <= "11111111111111111";  end process;  process  variable error\_status : boolean;  begin  wait on test\_in;  wait for 100ns;  if (( test\_in = "00000000000000000" and test\_out = '0') or  ( test\_in = "00000000000000001" and test\_out = '1') or  ( test\_in = "00000000000000010" and test\_out = '1') or  ( test\_in = "00000000000000011" and test\_out = '0') or  ( test\_in = "00000000000000100" and test\_out = '1') or  ( test\_in = "00000000000000101" and test\_out = '0') or  ----........  ( test\_in = "11111111111111100" and test\_out ='1') or  ( test\_in = "11111111111111101" and test\_out = '1') or  ( test\_in = "11111111111111110" and test\_out = '1') or  ( test\_in = "11111111111111111" and test\_out = '1'))  then  error\_status := false;  else  error\_status := true;  end if;  --error reporting  assert not error\_status  report "test failed."  severity note;  end process;  end tb\_arch ;  ---configuration  configuration demo-config of room\_automation\_testbench is  for tb\_arch  for uut:room\_automation  use entity work.room\_automation (sop\_arch);  end for;  end for;  end demo\_config; |
| --- |

**TRUTH TABLE :**

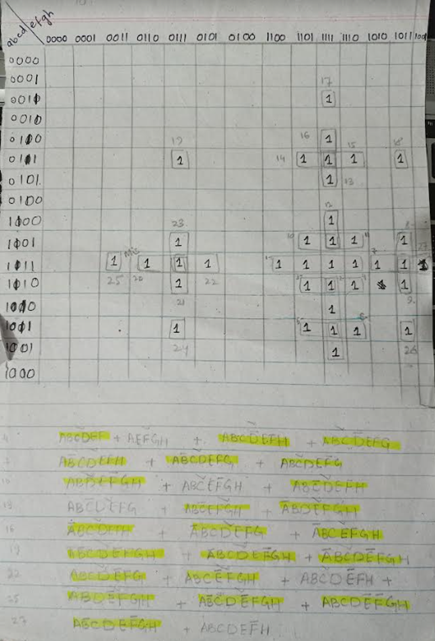
**FAN :**

****

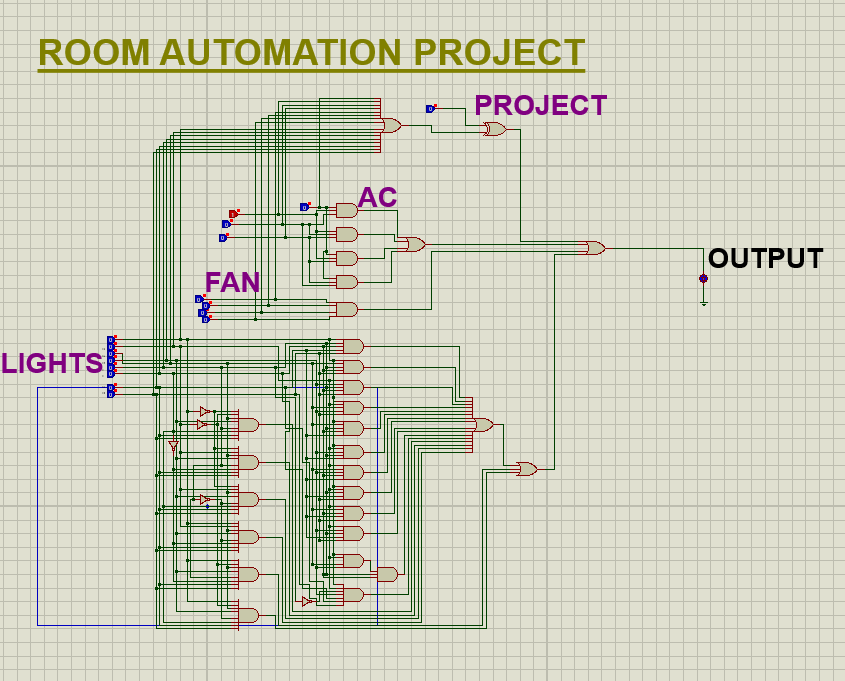
**AC:**

****

**LIGHTS :**

****

**SIMULATION AND CIRCUIT DIAGRAM :**

****

**CONCLUSION :**

This circuit can be made better by using fewer gates and more effective circuit design to combine the potential outputs. By putting the knowledge into practice on various logics, we may also adjust the type of gates.