Our Idea:

We implemented a smart watering system for garden plants. The self-watering system knows the best irrigation time based on the moisture and temperature levels that reach the plant, resulting in an efficient use of the water (saving water) and at the same time fulfilling all the needs of the plant.
We used Arduino Uno to convert the analog values coming from the soil and temperature sensors into digital bits, which were then sent to the FPGA board every specific unit time. The FPGA board controlled the operation of the water pump based on these values. Also, three 7 segment displays of the FPGA board were used to display the values of the moisture and temperature levels. Two digits (no. 0 and no.1) were used to display the temperature level in Centigrade. The third digit (no. 3) was used to display the moisture level from 0 to 7. (0 → High moisture || 7 → Low Moisture).

Note that throughout the project, we considered that the optimal conditions for the plant is temperature which is less than 35ºC and moisture level of less than 4.

Materials used:

1-FPGA DE-10 lite board

2-Breadboard

3-Arduino Uno microcontroller board

4- 6 VDC Water pump

5- 9 Volts Battery

6-Temperature sensor

7-Soil sensor

8-Jumper wires

9- Pump Hose

The Code:

First: Arduino code

Initializing the variables in use

```
const int tempsensor=A5; // Assigning analog pin A5 to variable Temprature 'sensor'
int tempc=0; ; //variable to store temperature in degree Celsius
float vout; //temporary variable to hold sensor reading
int temp_out1_0=5; //Pin that will output a digital value for the least Significant Bit for temperature unit digit
int temp_out1_1=6; //Pin that will output a digital value for the second Bit for temperature unit digit
int temp_out1_2=7; //Pin that will output a digital value for the third Bit for temperature unit digit
int temp_out1_3=8; //Pin that will output a digital value for the most  Significant Bit for temperature unit digit

int temp_out2_0=9; //Pin that will output a digital value for the least Significant Bit for temperature tenth digit
int temp_out2_1=10; //Pin that will output a digital value for the second Significant Bit for temperature tenth digit
int temp_out2_2=11; //Pin that will output a digital value for the third Significant Bit for temperature tenth digit
int temp_out2_3=12; //Pin that will output a digital value for the most Significant Bit for temperature tenth digit

int templeft; //tempoerary variables
int tempright;


int Moisture_Level;
int Moist_out_0 = 2;//Pin that will output a digital value for the Most Significant Bit for Moisture
int Moist_out_1 = 3;//Pin that will output a digital value for the second bit for Moisture
int Moist_out_2 = 4;//Pin that will output a digital value for the Least Significant Bit for Moisture
```

Configuring input and output pins

```
void setup() {

  pinMode(tempsensor,INPUT); // Configuring sensor pin as input
  pinMode(temp_out1_0, OUTPUT);
  pinMode(temp_out1_1, OUTPUT);
  pinMode(temp_out1_2, OUTPUT);
  pinMode(temp_out1_3, OUTPUT);

  pinMode(temp_out2_0, OUTPUT);
  pinMode(temp_out2_1, OUTPUT);
  pinMode(temp_out2_2, OUTPUT);
  pinMode(temp_out2_3, OUTPUT);

  pinMode(Moist_out_0, OUTPUT);
  pinMode(Moist_out_1, OUTPUT);
  pinMode(Moist_out_2, OUTPUT);

  Serial.begin(9600);
```

```
void loop() {
  // read the input on analog pin 0:
  int MoistureValue = analogRead(A3);
  Moisture_Level = MoistureValue/128;
  // print out the value you read:
  Serial.print("Moisture: ");
  Serial.print(MoistureValue);
  digitalWrite(Moist_out_0, Moisture_Level %2)
  Moisture_Level = Moisture_Level/2;
  digitalWrite(Moist_out_1, Moisture_Level %2)
  Moisture_Level = Moisture_Level/2;
  digitalWrite(Moist_out_2, Moisture_Level %2)

  Serial.print("=");
  Serial.print(Moist_out_2);
  Serial.print(Moist_out_1);
  Serial.print(Moist_out_0);

  vout=analogRead(tempsensor); //Reading the v

  vout=(vout*500)/1023;

  tempc=vout; // Storing value in Degree Celsi

  // ASSIGNING BINARY VALUES
  tempright = tempc%10;
  templeft= tempc/10;

  digitalWrite(temp_out1_0, tempright %2);
  tempright = tempright/2;
  digitalWrite(temp_out1_1, tempright %2);
  tempright = tempright/2;
  digitalWrite(temp_out1_2, tempright %2);
  tempright = tempright/2;
  digitalWrite(temp_out1_3, tempright %2);


   digitalWrite(temp_out2_0, templeft %2);
   templeft = templeft/2;
   digitalWrite(temp_out2_1, templeft %2);
   templeft = templeft/2;
   digitalWrite(temp_out2_2, templeft %2);
   templeft = templeft/2;
   digitalWrite(temp_out2_3, templeft %2);


   // assigning binary values

   Serial.print("| DegreeC=");
   Serial.print(tempc);
   Serial.print("=");
   Serial.print(temp_out2_3);
   Serial.print(temp_out2_2);
   Serial.print(temp_out2_1);
   Serial.print(temp_out2_0);
   Serial.print(" ");
   Serial.print(temp_out1_3);
   Serial.print(temp_out1_2);
   Serial.print(temp_out1_1);
   Serial.print(temp_out1_0);



   Serial.println();



   delay(10000);        // check every 30 mins
  }
```

The loop that reads the values of the sensors, converts temperature into Centigrade, converts both temperature and moisture readings into binary, and finally outputs them into pins which are connected to the FPGA board input.

Second: VHDL Code

Libraries in use and Entity

```vhdl

ENTITY MyProject IS
PORT (
 Temp0,Temp1: In std_logic_vector(3 downto 0);                --The two temperature digits (each in binary)
 Moisture:In std_logic_vector(2 downto 0);                    --The moisture level (in bianary)
 Seg7_1, Seg7_2, Seg7_4 : Buffer std_logic_vector(6 downto 0); --To display the readings on the segment displays
 Operate : Out std_logic;                                     -- Our Output that "operates" the pump
 O_Led : Out std_logic      )  ;                              -- A Led to display the output for debugging
 END MyProject;

```

Starting Architecture and displaying temperature unit digit on the first 7 segment display

```vhdl
ARCHITECTURE Seg7 Of MyProject IS

BEGIN
process(Temp0)

--Temperature unit digit display on first 7 segment display
begin

  case Temp0 is
   when "0000" => Seg7_1 <= "1000000"; ---0
   when "0001" => Seg7_1 <= "1111001";
   when "0010" => Seg7_1 <= "0100100";
   when "0011" => Seg7_1 <= "0110000";
   when "0100" => Seg7_1 <= "0011001";
   when "0101" => Seg7_1 <= "0010010";
   when "0110" => Seg7_1 <= "0000010";
   when "0111" => Seg7_1 <= "1111000";
   when "1000" => Seg7_1 <= "0000000";
   when "1001" => Seg7_1 <= "0011000";
   when Others => Seg7_1 <= "1111111";
  end case;
end process;

```

Displaying temperature tenth digit on the second 7 segment display

```
  --Temperature tenth digit display on second 7 segment display

  process(Temp1)

  begin
    case Temp1 is
     when "0000" => Seg7_2 <= "1000000"; ---0
     when "0001" => Seg7_2 <= "1111001";
     when "0010" => Seg7_2 <= "0100100";
     when "0011" => Seg7_2 <= "0110000";
     when "0100" => Seg7_2 <= "0011001";
     when "0101" => Seg7_2 <= "0010010";
     when "0110" => Seg7_2 <= "0000010";
     when "0111" => Seg7_2 <= "1111000";
     when "1000" => Seg7_2 <= "0000000";
     when "1001" => Seg7_2 <= "0011000";
     when Others => Seg7_2 <= "1111111";

    end case;
end process;
```

Displaying Moisture level (0-7) on the fourth 7 segment display

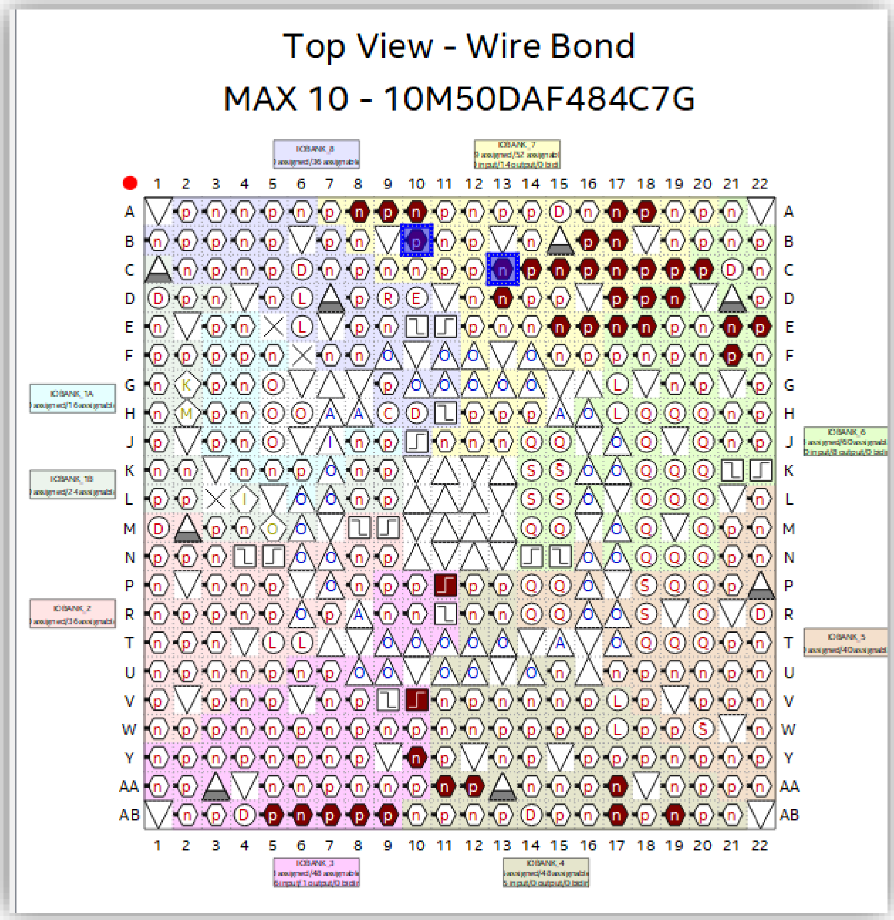
```

  --Moisture level (0-7) display on fourth 7 segment display

  process(Moisture)

  begin
    case Moisture is
     when "000" => Seg7_4 <= "1000000"; ---0
     when "001" => Seg7_4 <= "1111001";
     when "010" => Seg7_4 <= "0100100";
     when "011" => Seg7_4 <= "0110000";
     when "100" => Seg7_4 <= "0011001";
     when "101" => Seg7_4 <= "0010010";
     when "110" => Seg7_4 <= "0000010";
     when "111" => Seg7_4 <= "1111000";
     when Others => Seg7_4 <= "1111111";

    end case;
end process;
```

Adjusting the output that operates the pump based on the temperature and moisture levels and then, ending the architecture

```
process(Moisture,Temp0,Temp1)
begin

if(Moisture>"100" or (Temp0>"0101" and Temp1>"0011") or (Temp1>"0100") ) then
Operate<='1' ;
O_Led<='1';
else
Operate<='0';
O_Led<='0';
End if;

end process;



END Seg7;
```

Third: Pin Assignment

Moisture[2]	Input	PIN_AB7	3	B3_N0	PIN_AB7	2.5 V
Moisture[1]	Input	PIN_AB6	3	B3_N0	PIN_AB6	2.5 V
Moisture[0]	Input	PIN_AB5	3	B3_N0	PIN_AB5	2.5 V
O_Led	Output	PIN_A8	7	B7_N0	PIN_A8	2.5 V
Operate	Output	PIN_V10	3	B3_N0	PIN_V10	2.5 V
Seg7_1[6]	Output	PIN_C17	7	B7_N0	PIN_C17	2.5 V
Seg7_1[5]	Output	PIN_D17	7	B7_N0	PIN_D17	2.5 V
Seg7_1[4]	Output	PIN_E16	7	B7_N0	PIN_E16	2.5 V
Seg7_1[3]	Output	PIN_C16	7	B7_N0	PIN_C16	2.5 V
Seg7_1[2]	Output	PIN_C15	7	B7_N0	PIN_C15	2.5 V
Seg7_1[1]	Output	PIN_E15	7	B7_N0	PIN_E15	2.5 V
Seg7_1[0]	Output	PIN_C14	7	B7_N0	PIN_C14	2.5 V
Seg7_2[6]	Output	PIN_B17	7	B7_N0	PIN_B17	2.5 V
Seg7_2[5]	Output	PIN_A18	7	B7_N0	PIN_A18	2.5 V
Seg7_2[4]	Output	PIN_A17	7	B7_N0	PIN_A17	2.5 V
Seg7_2[3]	Output	PIN_B16	7	B7_N0	PIN_B16	2.5 V
Seg7_2[2]	Output	PIN_E18	6	B6_N0	PIN_E18	2.5 V
Seg7_2[1]	Output	PIN_D18	6	B6_N0	PIN_D18	2.5 V
Seg7_2[0]	Output	PIN_C18	7	B7_N0	PIN_C18	2.5 V
Seg7_4[6]	Output	PIN_E17	6	B6_N0	PIN_E17	2.5 V
Seg7_4[5]	Output	PIN_D19	6	B6_N0	PIN_D19	2.5 V
Seg7_4[4]	Output	PIN_C20	6	B6_N0	PIN_C20	2.5 V
Seg7_4[3]	Output	PIN_C19	7	B7_N0	PIN_C19	2.5 V
Seg7_4[2]	Output	PIN_E21	6	B6_N0	PIN_E21	2.5 V
Seg7_4[1]	Output	PIN_E22	6	B6_N0	PIN_E22	2.5 V
Seg7_4[0]	Output	PIN_F21	6	B6_N0	PIN_F21	2.5 V
Temp0[3]	Input	PIN_AA11	4	B4_N0	PIN_AA11	2.5 V
Temp0[2]	Input	PIN_Y10	3	B3_N0	PIN_Y10	2.5 V
Temp0[1]	Input	PIN_AB9	3	B3_N0	PIN_AB9	2.5 V
Temp0[0]	Input	PIN_AB8	3	B3_N0	PIN_AB8	2.5 V
Temp1[3]	Input	PIN_AB19	4	B4_N0	PIN_AB19	2.5 V
Temp1[2]	Input	PIN_AA17	4	B4_N0	PIN_AA17	2.5 V
Temp1[1]	Input	PIN_AB17	4	B4_N0	PIN_AB17	2.5 V
Temp1[0]	Input	PIN_AA12	4	B4_N0	PIN_AA12	2.5 V


Top View - Wire Bond
MAX 10 - 10M50DAF484C7G

Results

Case 1: The temperature reading was 37 and the moisture reading was 6.

Result 1: The pump operated.

Case 2: The temperature reading was 25 and the moisture reading was 6.

Result 2: The pump operated.

Case 3: The temperature reading was 39 and the moisture reading was 3.

Result 3: The pump operated.

Case 3: The temperature reading was 26 and the moisture reading was 1.

Result 3: The pump did not operate.

Conclusion: The pump successfully operated when any of the temperature and moisture (or both) were not optimal. The pump did not operate when they were optimal.