DEPARTMENT OF ELECTRONICS & COMM. ENGINEERING,
NIT WARANGAL

DSD MINI PROJECT

FUZZY LOGIC IMPLEMENTATION IN VHDL:
"Temperature controller on FPGA"

Submitted By:

Marisetti RVVS SaiTeja	164236
Menneni sai krishna Soumith	164237
Mettu Bhargavi	164238
NC Ashish	164239

Guided By:

Shri Naresh Kumar

Department Of Electronics And Communication Engineering

This is to certify that undersigned students of II/IV B.Tech , Section-B Electronics and Communications Engineering Branch, have completed their mini project entitled "**Temperature controller on FPGA using Fuzzy Logic** " for the partial fulfilment of Digital System and Designing-II laboratory.

Name	Roll No	Signature
Marisetti RVVS SaiTeja	164236	----------
Menneni saikrishna Soumith	164237	----------
Mettu Bhargavi	164238	----------
NC Ashish	164239	----------

Signature of the Faculty

-Shri Naresh kumar
Department of ECE

INDEX

TOPIC	PAGE NUMBER
TITLE	1
CERTIFICATE	2
INDEX	3-4
PROBLEM STATEMENT	5
ABSTRACT	5
INTRODUCTION	6
PRIOR WORK	6
TOP MODULES	9-12
PROPOSED APPLICATION	13
VHDL CODE FOR TEMPERATURE FUZZIFIER	22
SIMULATION RESULTS FOR TEMPERATURE FUZZIFIER	26
VHDL CODE FOR TEMP RATE FUZZIFIER	27
SIMULATION RESULTS FOR TEMP RATE FUZZIFIER	30

VHDL CODE FOR FUZZY RULE MATRIX 31

SIMULATION RESULTS FOR FUZZY RULE MATRIX 33

VHDL CODE FOR DEFUZZIFIER 34

SIMULATION RESULTS FOR DEFUZZIFIER 36

VHDL CODE FOR
FUZZY LOGIC CONTROLLER 37

SIMULATION RESULTS FOR FLC 40

CONCLUSION 41

PROBLEM STATEMENT:

DESIGN STRUCTURAL VHDL MODEL FOR TEMPERATURE CONTROLLER USING FUZZY LOGIC AND SIMULATE TO VERIFY THE FUNCTIONALITY AND SYNTHESIS TO VIEW RTL SCHEMATIC.

ABSTRACT :-

Our project aim is to implement fuzzy logic based temperature controller(on AC or Fan) using vhdl . Fuzzy controller designs have become valuable in large number of applications. There are three basic steps for fuzzy logic controller i.e fuzzification of inputs, defuzzification of outputs, and rule base. In this designed temperature controller there are 2-inputs, 1-output and 5-membership functions. According to the 2 inputs(temperature and rate of change of temperature), the output (speed of air conditioner) is controlled based on relation between the fuzzy set (cold,cool,mild,warm,hot,slow,moderate,fast) .The purpose of this application is to control the speed .

PROPOSED APPLICATION:-*Temperature controller based on fuzzy logic*
Prior work:-Car Parking System
Used Logic:-Fuzzy logic

INTRODUCTION:-

The word “fuzzy” is defined as “indistinct, imprecisely defined”. Fuzzy systems are systems to be precisely defined. Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic.Logics based on the concept of fuzzy sets, in which membership is expressed in varying probabilities or degrees of truth—that is, as a membership of values ranging from 0 (does not occur) to 1 (definitely occurs).
In this paper, the implementation of FLC will be done by VHDL which is the most important platform for hardware implementation due to low power and high operational speed.

PRIOR WORK:-

***FPGA based autonomous parking of a car like robot using Fuzzy Logic Control.

1)Design of Low power VLSI-Implementation of Fuzzy Logic based automatic car parking system:

-->Proposed architecture of this system consists of two main components.
1.FLC(FUZZY LOGIC CONTROL) to determine speed as well as direction of car movement while parking.

2.FSM(FINAL STATE MACHINE) that controls operating system of car.Both these blocks operates jointly which changes state based on data obtained from 6 sensors those are integrated on front,rear,right and left sides of car.

*Disadvantage:*In architecture of Defuzzifier,more number of multipliers,comparators,adders are used,i.e half of the chip area is consumed by FLC and Power consumption increases Rapidly.

2)Prototype of an under ground Multistoried Automated Car Parking System:

-->Multistorey Car parking is becoming increasingly popular as they enable to conserve space.

-->parking on multiple floors brings the need of using elevating mechanism for lifting the vehicles.

-->Control systems are used for coordination between the vehicle and lift mechanism and vacant parking space detection .

-->The whole operation of system is automated by software programmed in to micro-controller.

Disadvantages: The initial cost of building an underground parking system is very high.

There will be waiting problem for both parking and retrieval of car as the system uses singular shaft.

3)Fuzzy Logic Control of autonomous vehicles for parallel parking :

-->Automatic path tracking approach has great deal of attention,where we solve this by skill based approach i.e ,Fuzzy logic control networks are used.

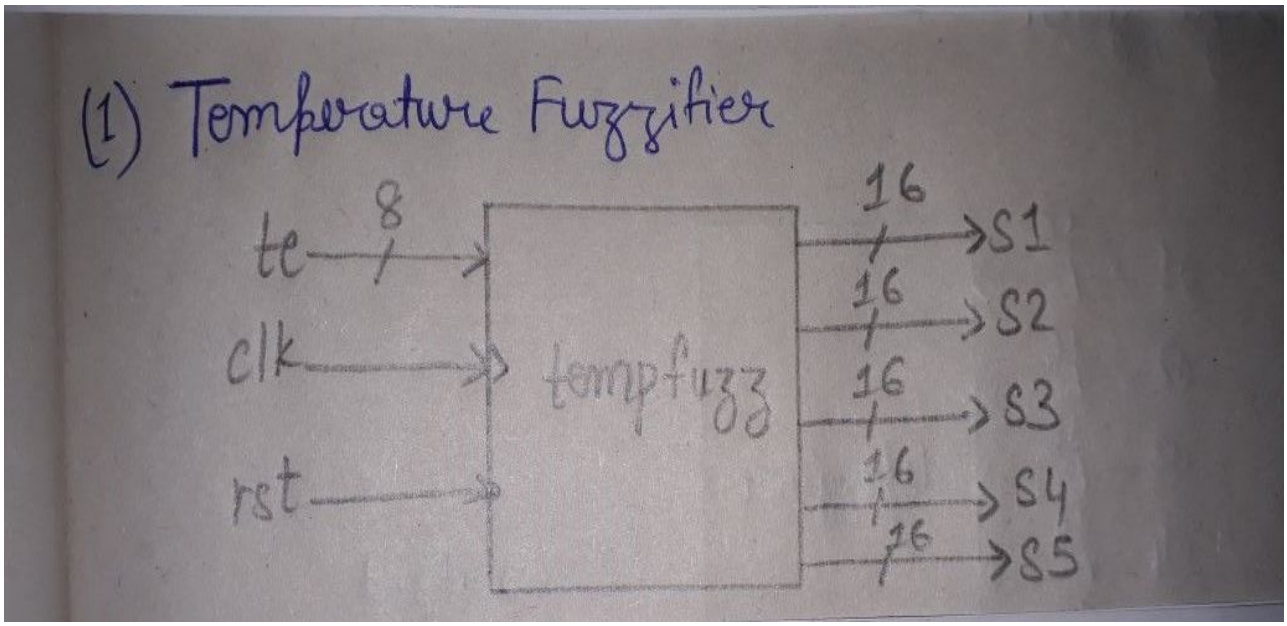
-->In FLC<the steering angle was generated based on longitudinal and lateral distance of vehicle to parking space and orientation of the vehicle .

-->Describes the automated parking process and develops fuzzy logic controllers.

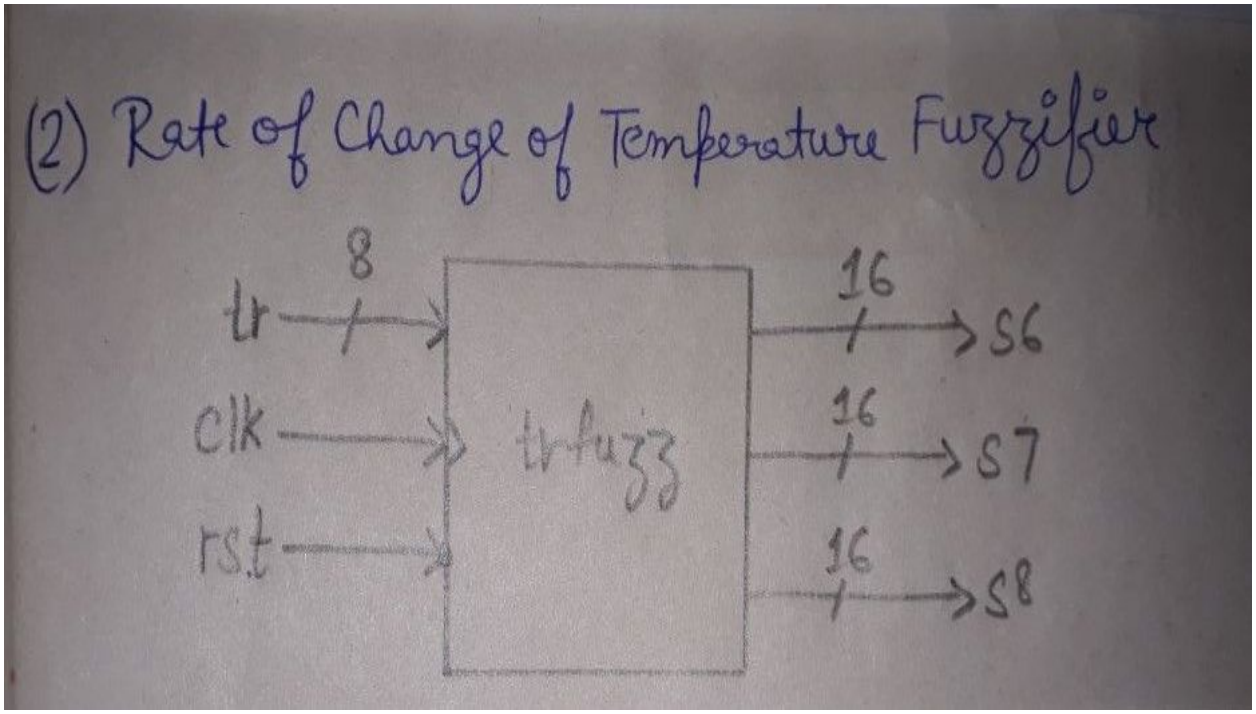
-->Simulation and experimental results illustrate the effectiveness of parallel parking scheme to move the vehicle.

*Disadvantage:*In architecture of Defuzzifier,more number of multipliers,comparators,adders are used,i.e half of the chip area is consumed by FLC and Power consumption increases rapidly. 08

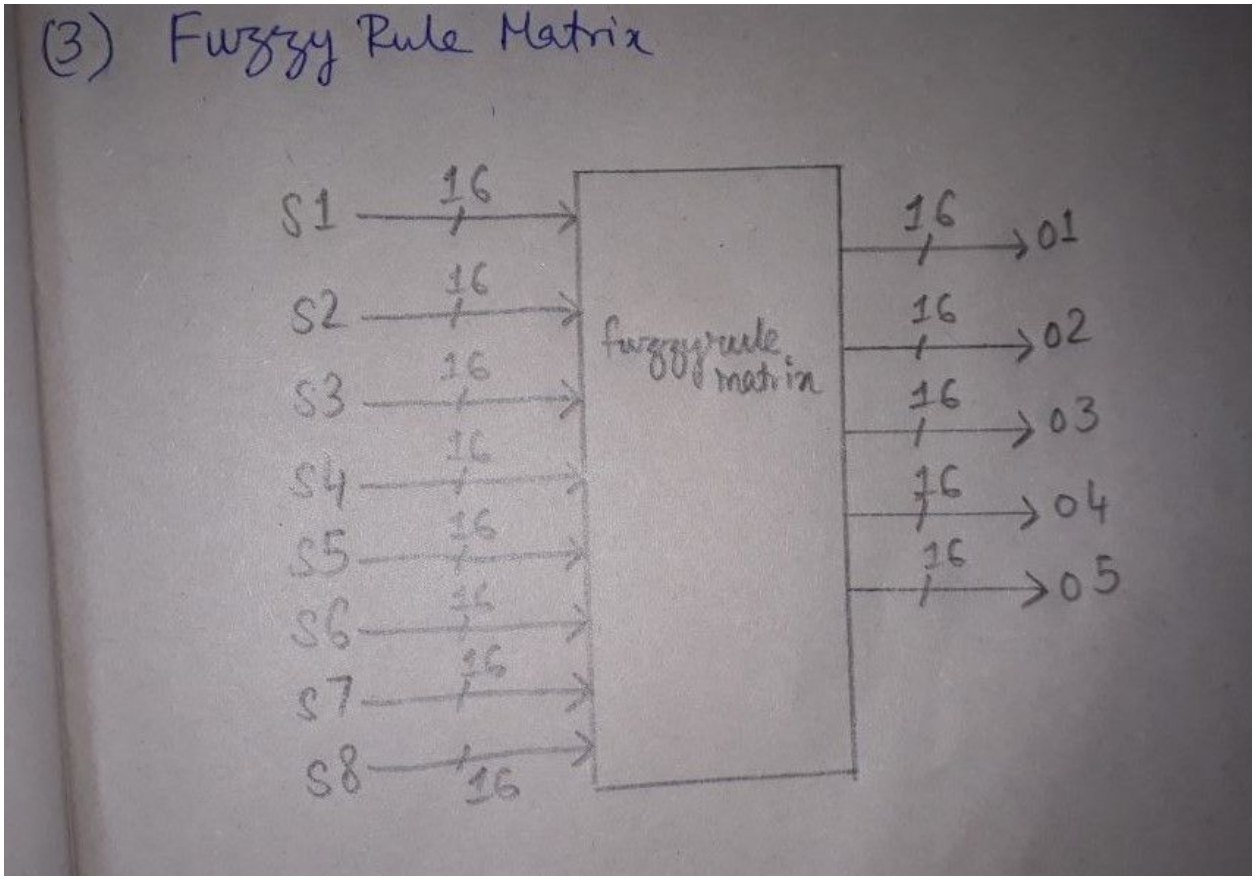
TOP MODULES



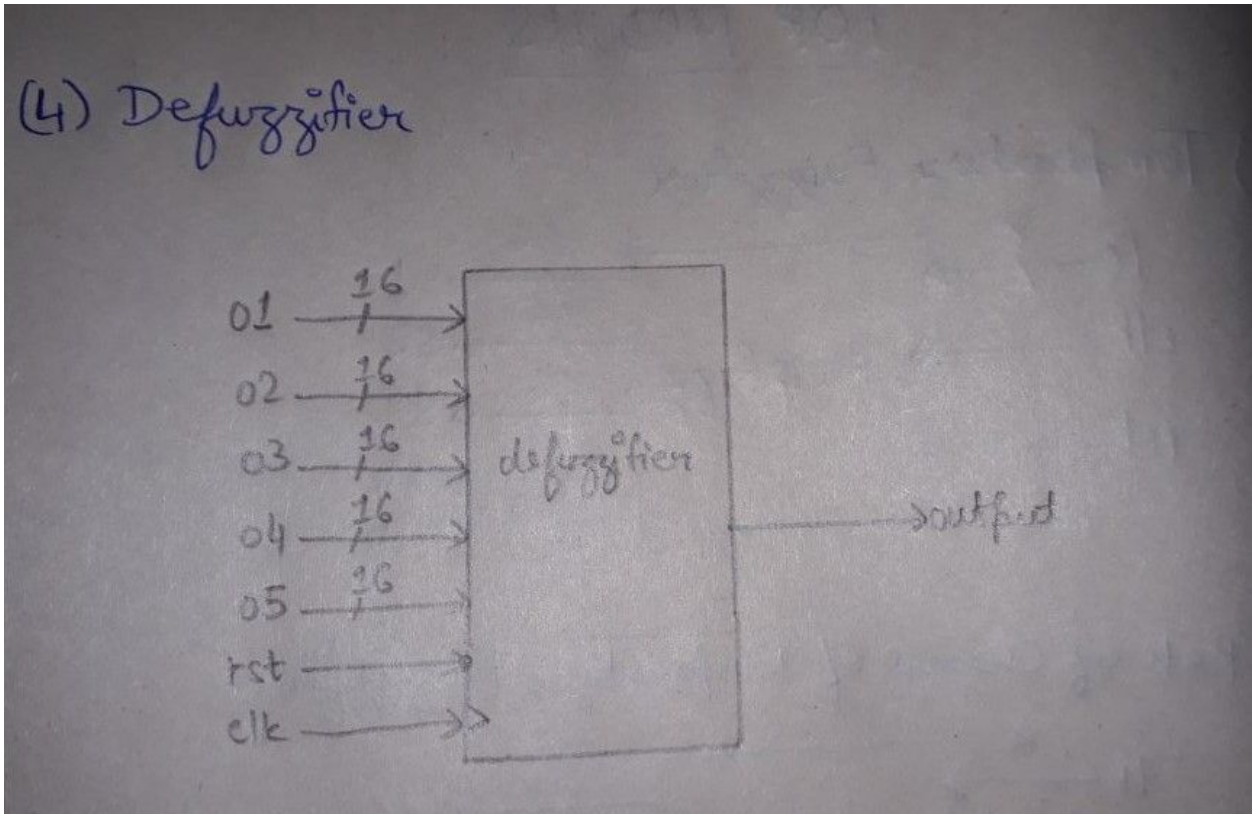
a)TOP MODULE FOR TEMPERATURE FUZZIFIER



b)TOP MODULE FOR RATE OF CHANGE OF TEMPERATURE FUZZIFIER



C)TOP MODULE FOR FUZZY RULE MATRIX



D)TOP MODULE FOR DEFUZZIFIER

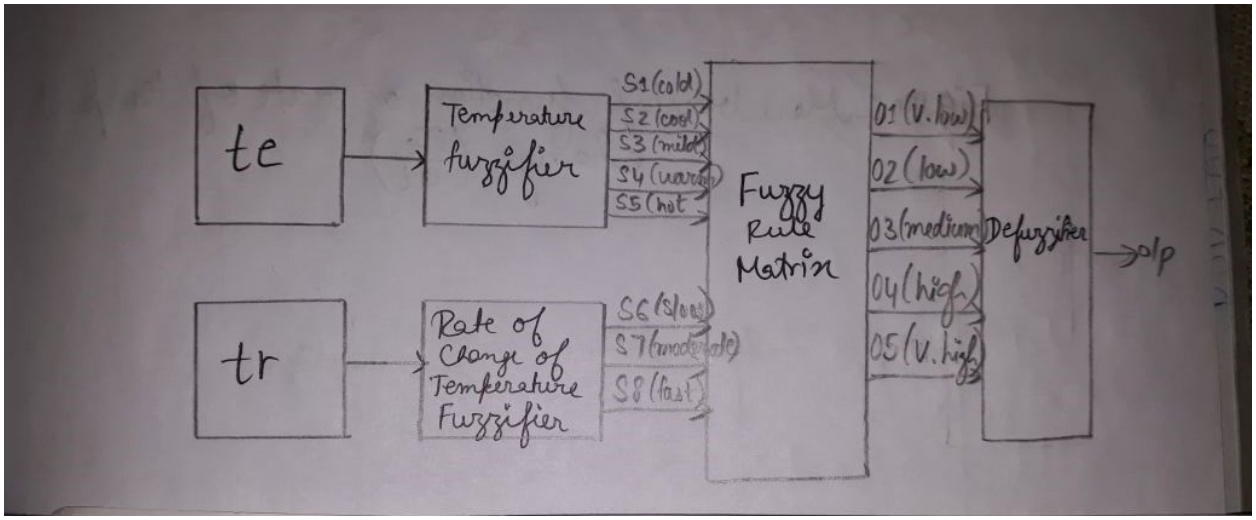
PROPOSED APPLICATION:

FUZZY LOGIC CONTROLLER MODULE

Fuzzy control system design essentially amounts to:
1. Choosing the fuzzy controller inputs, outputs and their membership degrees.
2. Choosing the pre-processing that is needed for the controller inputs and possibly post processing that is needed for the output.
3. Designing each of the four components of fuzzy controller.
The fuzzy controller takes input values. These values are referred to as "crisp" since they are represented as a single number, not a fuzzy one. In order for the fuzzy controller to understand the inputs, the crisp input has to be converted to a fuzzy number. This process is called ***fuzzification***. The next step in the fuzzy control process is the ***implementation of the rule base***. This is where the fuzzy inputs are compared and on the membership of each, the fuzzy output is chosen.

A system is designed for implementation of temperature controller using VHDL using two input variables with standard membership functions fuzzifier as 'Temperature fuzzifier' with membership functions cold,cool,mild,warm,hot and 'Rate of change of Temperature' with membership functions slow,medium and fast.

Fuzzy rule matrix is used to interface these two inputs into an output variable that describes the speed of temperature controller with five membership functions very slow,slow,medium,high and very high.



STRUCTURAL VIEW OF SYSTEM

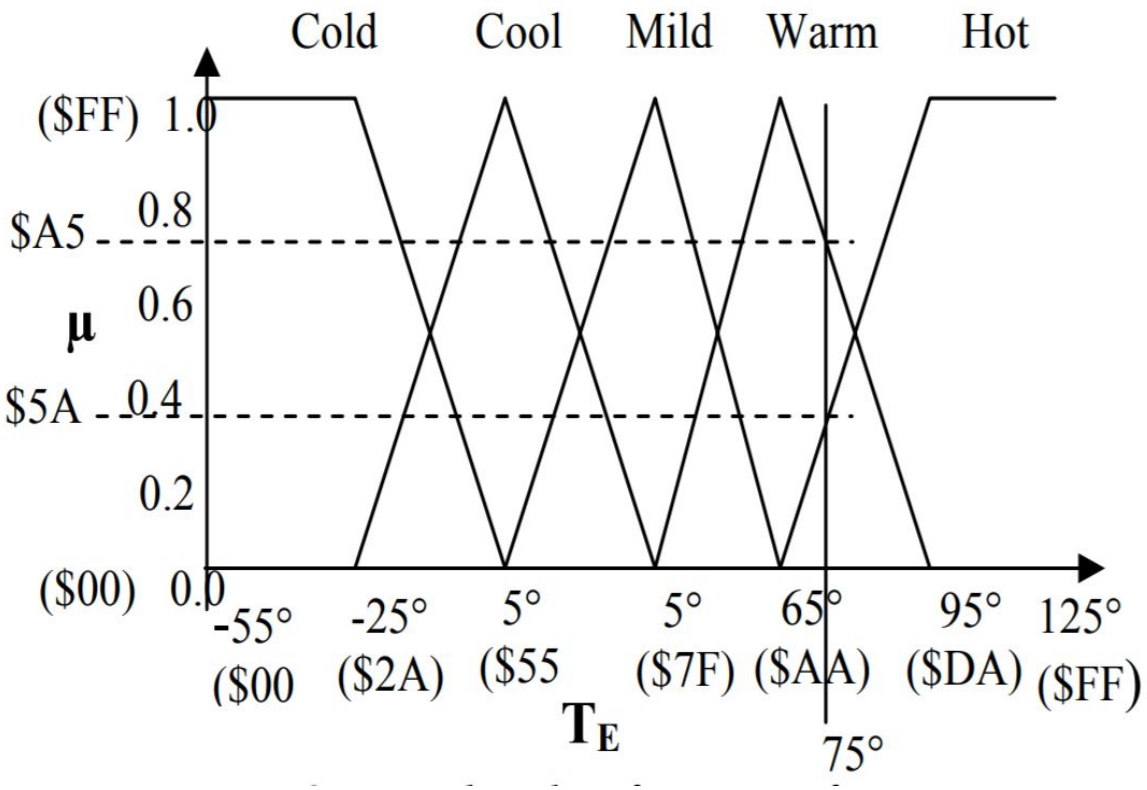


FIG.MEMBERSHIP FUNCTION OF TEMPERATURE

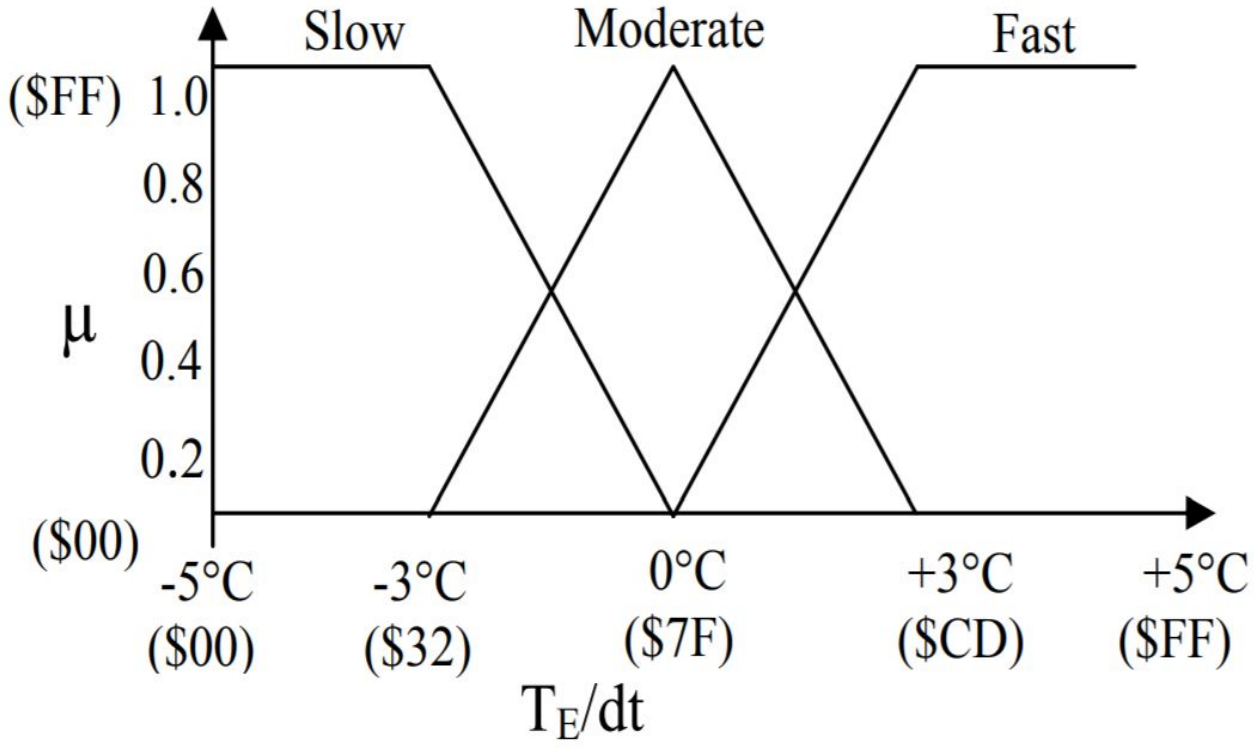
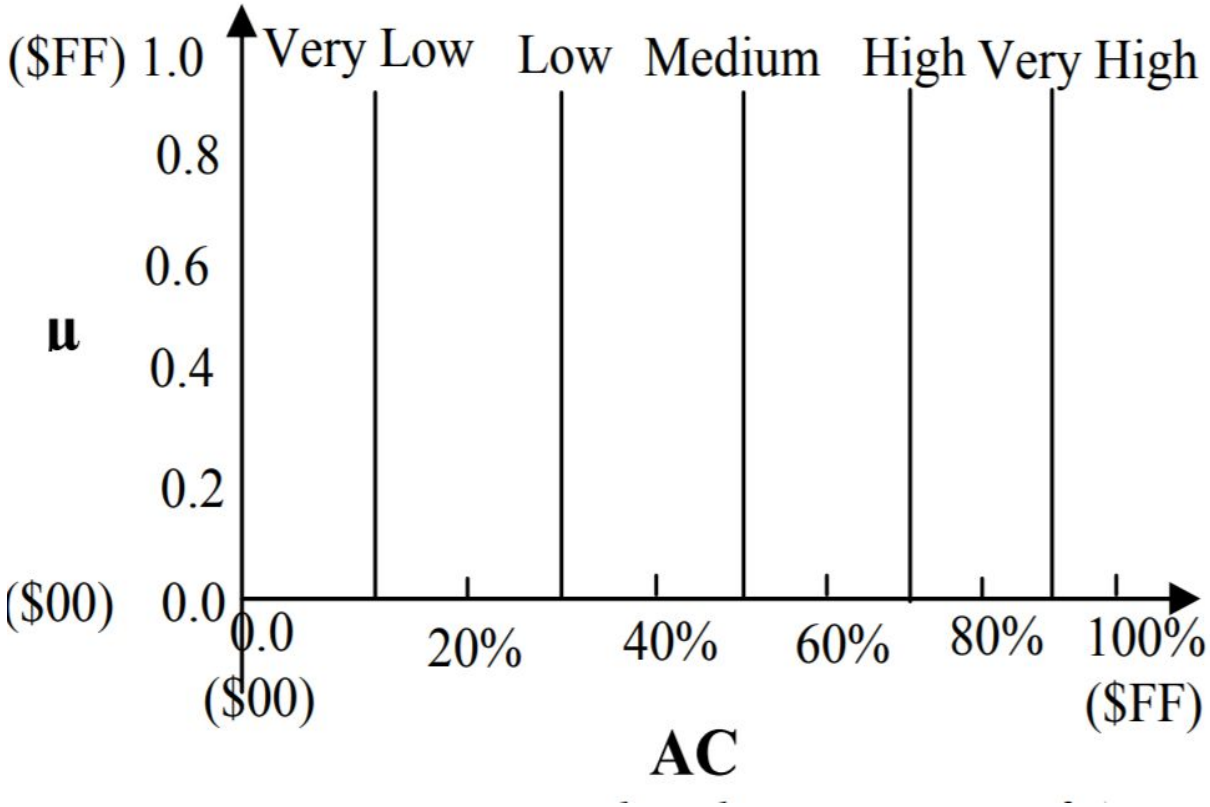


FIG.MEMBERSHIP FUNCTION OF RATE OF TEMPERATURE CHANGE



OUTPUT MEMBERSHIP FUNCTION

If-then rules

The main part of the fuzzy controller is the rule base as it would represent the more detailed information about how to regulate the speed of the controller.

If-then rule statements are used to formulate the conditional statements. Fuzzy sets provide an ultimate mechanism of communication between humans and computing environment.
A single fuzzy if-then rule assumes the form as specified below

If X is A then Y is B

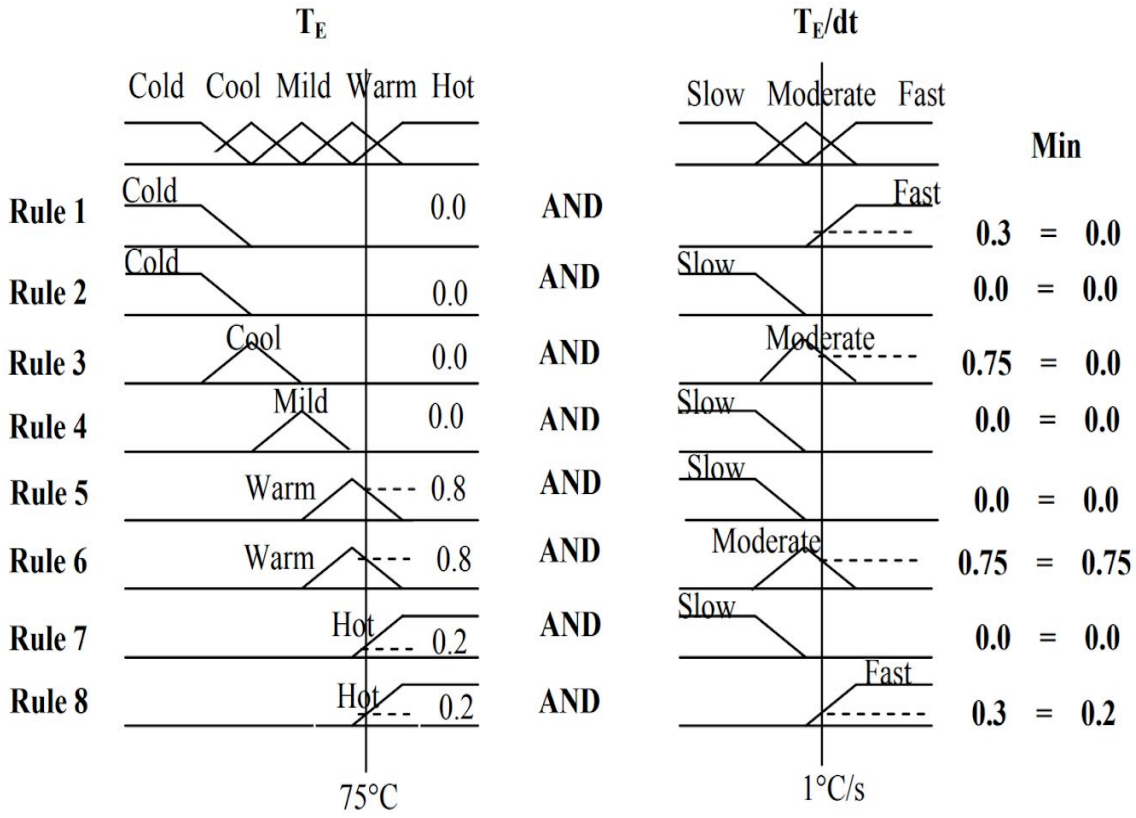
Where A and B are linguistic values defined by fuzzy sets on the ranges X and Y (Universes of discourse).

The if part of the rule “X is A” part is called the antecedent or premise, while the then part of the rule “Y is B” is called consequent or conclusion. The concept A is represented as a number between 0 and 1, and so the antecedent is an interpretation that results a single number between 0 and 1.

The primary objective of constructing fuzzy if-then rule matrix is to map out the universe of possible inputs while keeping the system sufficiently under control.

IF				THEN
Rule	**Temperature Error (T_E)**	**Logical Operator**	**Rate Change (T_E/dt)**	**Air Conditioning (AC)**
1	Cold	AND	Fast	Very High
2	Cold	AND	Slow	High
3	Cool	AND	Moderate	Medium
4	Mild	AND	Slow	Medium
5	Warm	AND	Slow	Low
6	Warm	AND	Moderate	Low
7	Hot	AND	Slow	Very Low
8	Hot	AND	Fast	Very Low

RULE EVALUATION TABLE



RULE EVALUATION EXAMPLE

Fuzzy if-then rules for the proposed model are:
INPUTS-temperature ,rate of change of temperature
OUTPUT-speed of air conditioner
Rule 1 : very high<= minimum(cold,fast)
Rule 2 : high<=minimum(cold,slow)
Rule 3 :
medium<=maximum(minimum(cold,moderate),minimum(mild,slow))
Rule 4 :
medium<=maximum(minimum(cold,moderate),minimum(mild,slow))
Rule 5
low<=maximum(minimum(warm,slow),minimum(warm,moderate))
Rule 6
low<=maximum(minimum(warm,slow),minimum(warm,moderate))
Rule 7
Very low<=maximum(minimum(hot,slow),minimum(hot,fast))
Rule 8
Very low<=maximum(minimum(hot,slow),minimum(hot,fast))

VHDL CODES

1)TEMPERATURE FUZZIFIER VHDL CODE

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity tempfuzz is
    Port ( te : in STD_LOGIC_VECTOR(7 DOWNTO 0);
           clk : in STD_LOGIC;
           rst :in std_logic;
           s1,s2,s3,s4,s5 : out STD_LOGIC_VECTOR (15 downto 0));
end tempfuzz;

architecture Behavioral of tempfuzz is
type cold is record
point1 :std_logic_vector(7 downto 0);
point2 :std_logic_vector(7 downto 0);
slope1 :std_logic_vector(7 downto 0);
slope2 :std_logic_vector(7 downto 0);
end record;
type cool is record
point1 :std_logic_vector(7 downto 0);
point2 :std_logic_vector(7 downto 0);
slope1 :std_logic_vector(7 downto 0);
slope2 :std_logic_vector(7 downto 0);
end record;
type mild is record
point1 :std_logic_vector(7 downto 0);
point2 :std_logic_vector(7 downto 0);
```

```
slope1 :std_logic_vector(7 downto 0);
slope2 :std_logic_vector(7 downto 0);
end record;
type warm is record
point1 :std_logic_vector(7 downto 0);
point2 :std_logic_vector(7 downto 0);
slope1 :std_logic_vector(7 downto 0);
slope2 :std_logic_vector(7 downto 0);
end record;
type hot is record
point1 :std_logic_vector(7 downto 0);
point2 :std_logic_vector(7 downto 0);
slope1 :std_logic_vector(7 downto 0);
slope2 :std_logic_vector(7 downto 0);
end record;


constant term_name : cold :=
(point1=>x"00",slope1=>"11111111",point2=>x"2a",slope2=>"0000011
0");
constant term_name1 : cool
:=(point1=>x"2a",slope1=>"00000110",point2=>x"55",slope2=>"00000
110");
constant term_name2 : mild
:=(point1=>x"55",slope1=>"00000110",point2=>x"7f",slope2=>"00000
110");
constant term_name3 : warm
:=(point1=>x"7f",slope1=>"00000110",point2=>x"aa",slope2=>"11111
111");
constant term_name4 : hot :=(point1=>
x"aa",slope1=>"11111111",point2=>x"da",slope2=>"11111111");
begin
process(clk)
begin
if rst='0' then
    s1<="0000000000000000";    s2<="0000000000000000";
s3<="0000000000000000";  s4<="0000000000000000";
```

```
s5<="0000000000000000";
    elsif clk='1' and clk'event then
if (te < term_name.point1 or te> term_name1.point2) then
s1<="0000000000000000";
elsif(te < term_name.point2) then
    s1<=(te-term_name.point1)*(term_name.slope1);
else
    s1<=255-(te-term_name.point2)*(term_name.slope2);
end if;
if (te < term_name1.point1 or te> term_name2.point2) then
s2<="0000000000000000";
elsif(te < term_name1.point2) then
    s2<=(te-term_name1.point1)*term_name1.slope1;
else
    s2<=255-(te-term_name1.point2)*term_name1.slope2;
end if;
if (te < term_name2.point1 or te> term_name3.point2) then
s3<="0000000000000000";
elsif(te < term_name2.point2) then
    s3<=(te-term_name2.point1)*term_name2.slope1;
else
    s3<=255-(te-term_name2.point2)*term_name2.slope2;
end if;
if (te < term_name3.point1 or te>term_name4.point2) then
s4<="0000000000000000";
elsif(te < term_name3.point2) then
    s4<=(te-term_name3.point1)*term_name3.slope1;
else
    s4<=255-(te-term_name3.point2)*term_name3.slope2;
end if;
if (te < term_name4.point1) then s5<="0000000000000000";
elsif(te < term_name4.point2) then
    s5<=(te-term_name4.point1)*term_name4.slope1;
else
    s5<=255-(te-term_name4.point2)*term_name4.slope2;
end if;
end if;
```

```
end process;

end Behavioral;
```

SIMULATION RESULT OF TEMPERATURE FUZZIFIER

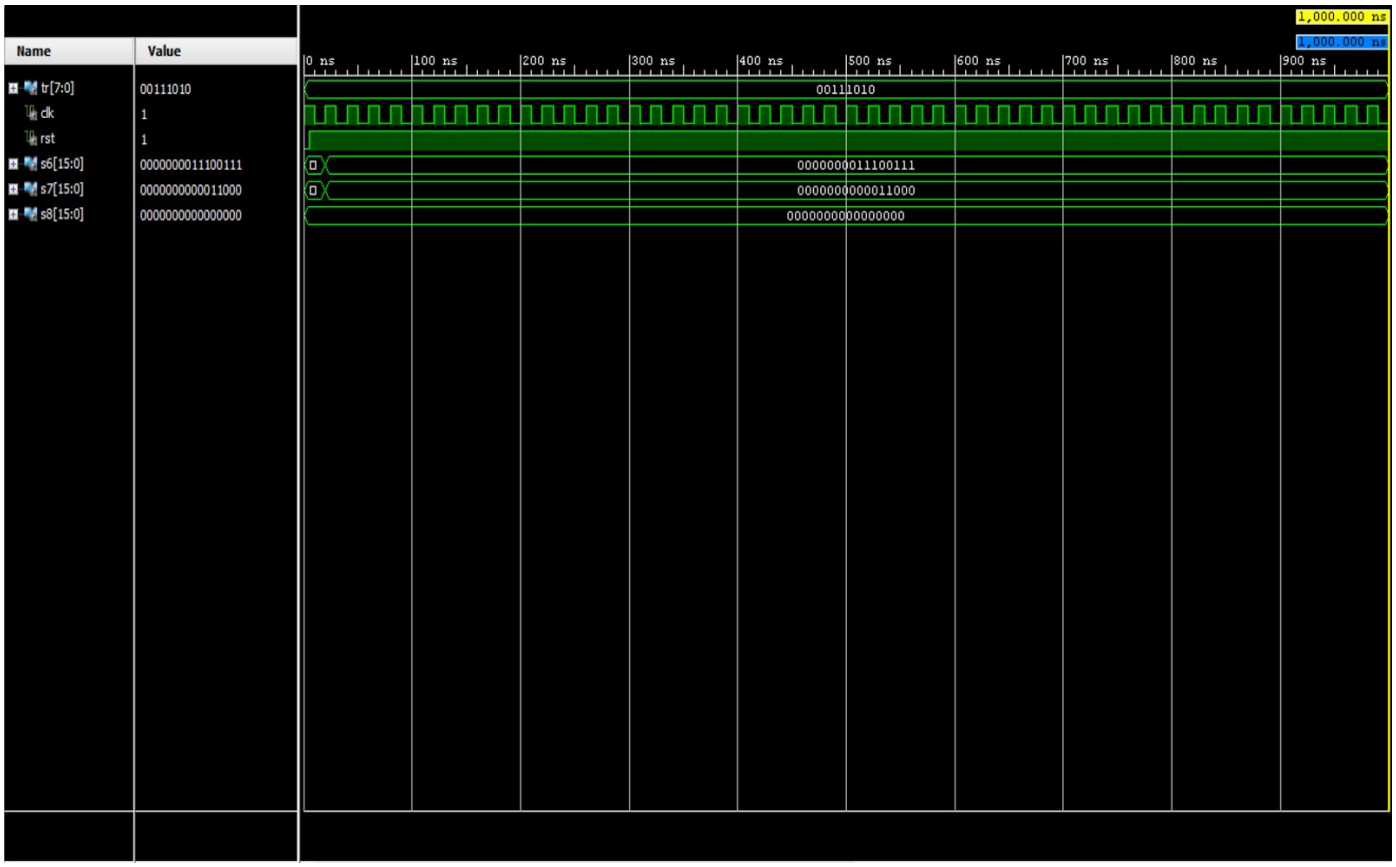
2)RATE OF CHANGE OF TEMPERATURE FUZZIFIER VHDL CODE

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.ALL;

entity trfuzz is
    Port ( tr : in STD_LOGIC_VECTOR(7 DOWNTO 0);
           clk : in STD_LOGIC;
           rst :in std_logic;
           s6,s7,s8 : out STD_LOGIC_VECTOR (15 downto 0));
end trfuzz;

architecture Behavioral of trfuzz is
type slow is record
point1 :std_logic_vector(7 downto 0);
point2 :std_logic_vector(7 downto 0);
slope1 :std_logic_vector(7 downto 0);
slope2 :std_logic_vector(7 downto 0);
end record;
type moderate is record
point1 :std_logic_vector(7 downto 0);
point2 :std_logic_vector(7 downto 0);
slope1 :std_logic_vector(7 downto 0);
slope2 :std_logic_vector(7 downto 0);
end record;
type fast is record
point1 :std_logic_vector(7 downto 0);
point2 :std_logic_vector(7 downto 0);
slope1 :std_logic_vector(7 downto 0);
slope2 :std_logic_vector(7 downto 0);
end record;
```

```
constant term: slow
:=(point1=>x"00",slope1=>"00000000",point2=>x"32",slope2=>"00000
011");
constant term1 : moderate
:=(point1=>x"32",slope1=>"00000011",point2=>x"7f",slop

e2=>"00000011");

constant term2 :
fast:=(point1=>x"7f",slope1=>"00000011",point2=>x"cd",slope2=>"1
1111111");
begin
process(clk)
begin
if rst='0' then
    s6<="0000000000000000";    s7<="0000000000000000";
s8<="0000000000000000";
    elsif clk='1' and clk'event then
if (tr < term.point1 or tr>term1.point2) then
s6<="0000000000000000";
elsif(tr < term.point2) then
    s6<=(tr-term.point1)*term.slope1;
else
    s6<=255-(tr-term.point2)*term.slope2;
end if;
if (tr < term1.point1 or tr>term2.point2) then
s7<="0000000000000000";
elsif(tr < term1.point2) then
    s7<=(tr-term1.point1)*term1.slope1;
else
    s7<=255-(tr-term1.point2)*term1.slope2;
end if;
if (tr < term2.point1) then s8<="0000000000000000";
elsif(tr < term2.point2) then
    s8<=(tr-term2.point1)*term2.slope1;
else
    s8<=255-(tr-term2.point2)*term2.slope2;
```

```
end if;
end if;
end process;
end Behavioral;
```

SIMULATION RESULT OF RATE OF CHANGE OF TEMPERATURE FUZZIFIER

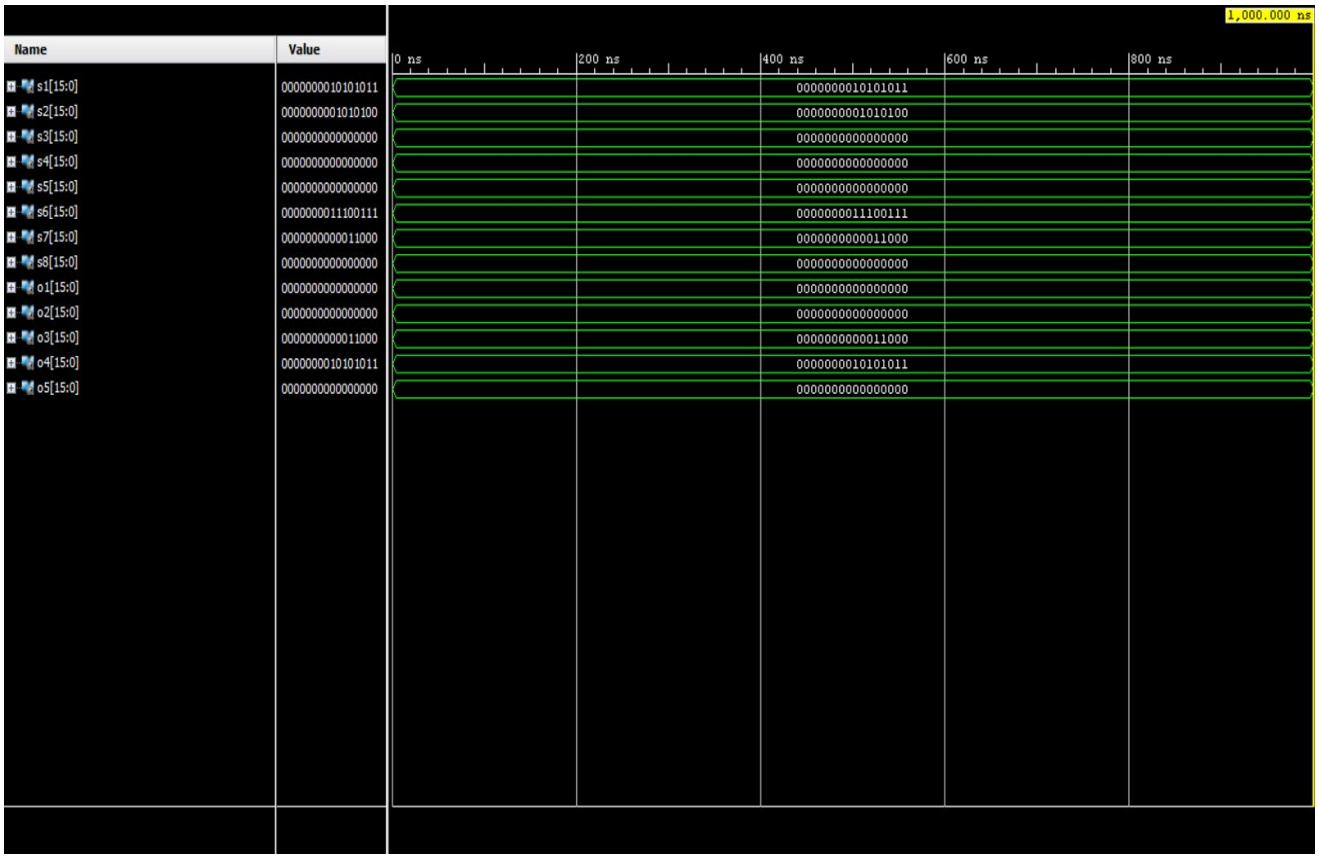


3)FUZZY RULE MATRIX VHDL CODE

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity fuzzyrulematrix is
    Port ( s1,s2,s3,s4,s5,s6,s7,s8: in STD_LOGIC_VECTOR (15
downto 0);
           o1,o2,o3,o4,o5 : out STD_LOGIC_VECTOR (15 downto 0));
end fuzzyrulematrix;

architecture Behavioral of fuzzyrulematrix is
signal verylow, low, medium,high, veryhigh :std_logic_vector(15
downto 0);

function minimum(a, b: std_logic_vector) return std_logic_vector
is
variable min: std_logic_vector(15 downto 0) := (others => '0');
begin if a < b then min := a; else min := b; end if; return min;
end minimum;

function maximum(a, b: std_logic_vector) return std_logic_vector
is
variable max: std_logic_vector(15 downto 0) := (others =>
'0');begin
if a > b then max := a; else max := b; end if; return max; end
maximum;

begin
rule1 : veryhigh <=minimum(s1,s8);
rule2 :high<=minimum(s1,s6);
rule3: medium<=maximum(minimum(s2,s7),minimum(s3,s6));
rule4 :medium<=maximum(minimum(s2,s7),minimum(s3,s6));
```

```
rule5 :low<=maximum(minimum(s4,s6),minimum(s4,s7));
rule6 :low<=maximum(minimum(s4,s6),minimum(s4,s7));
rule7 :verylow<=maximum(minimum(s5,s6),minimum(s5,s8));
rule8 :verylow<=maximum(minimum(s5,s6),minimum(s5,s8));
o5<=veryhigh;
o4<=high;
o3<=medium;
o2<=low;
o1<=verylow;
end Behavioral;
```

SIMULATION RESULT OF FUZZY RULE MATRIX



4)DEFUZZIFIER VHDL CODE

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;


entity defuzzifier is
Port ( o1,o2,o3,o4,o5 : in STD_LOGIC_VECTOR (15 downto
0);
rst ,clk:in std_logic;
output : out STD_LOGIC_vector(31 downto 0));
end entity;

architecture Behavioral of defuzzifier is
signal t,h,q:integer;
begin
process(clk)
variable o,x,y,z,u,v,p,s,m:integer;
begin
x:=conv_integer(o1);
y:=conv_integer(o2);
z:=conv_integer(o3);
u:=conv_integer(o4);
v:=conv_integer(o5);
p:=42*x+85*y+127*z+168*u+210*v;
s:=x+y+z+u+v;t<=p;h<=s;
o:=0;
m:=p;
```

```
for i in 0 to m loop
    p:=p-s;
    o:=o+1;
    if p<s then
    exit;
    end if;
    end loop;
q<=o;
output<=conv_std_logic_vector(o,32);
end process ;
end Behavioral;
```

SIMULATION RESULT OF DEFUZZIFIER

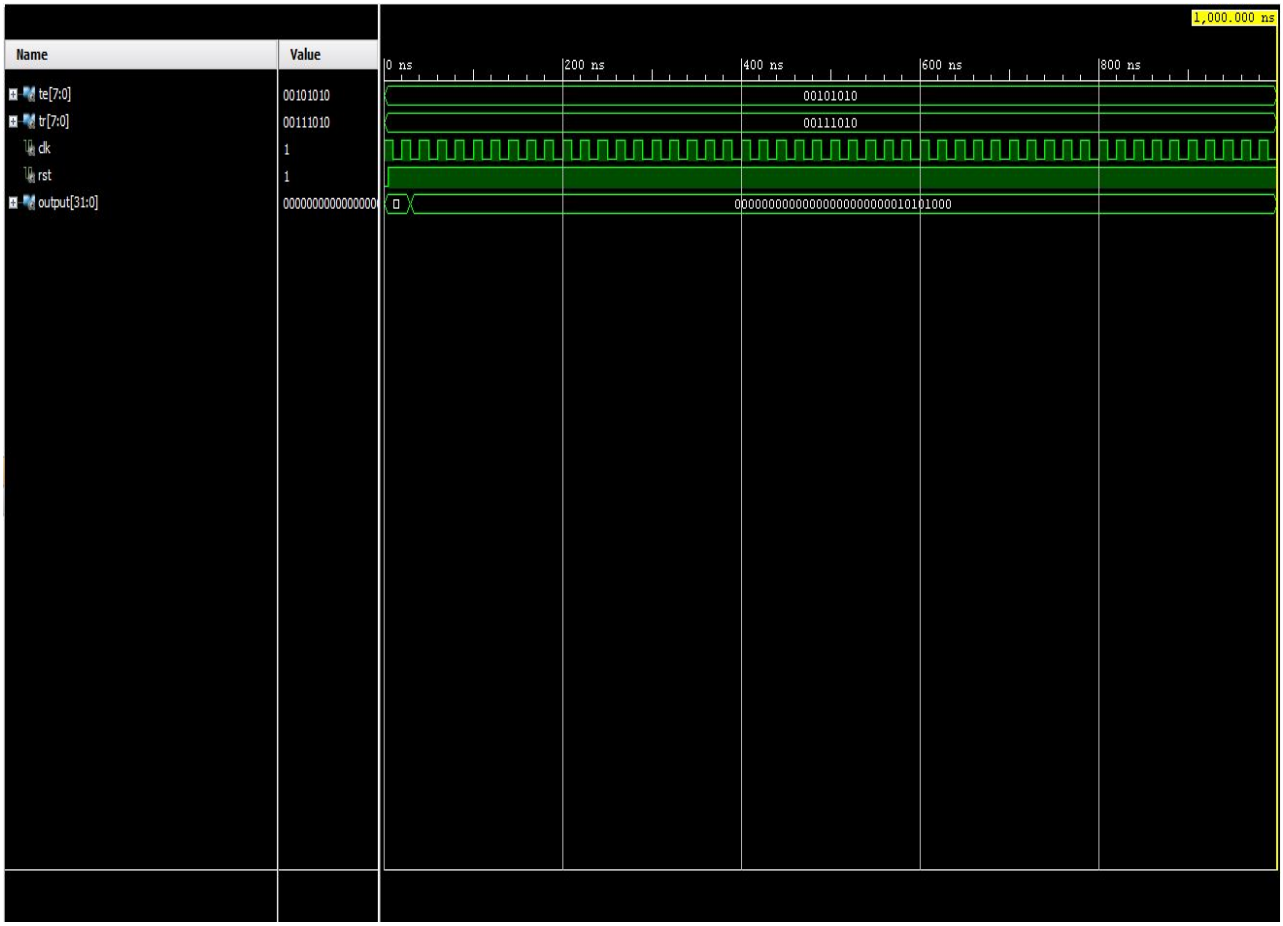
5)FUZZY LOGIC CONTROLLER VHDL CODE

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;


entity flc is
    Port ( te : in STD_LOGIC_VECTOR(7 DOWNTO 0);
           tr : in STD_LOGIC_VECTOR(7 DOWNTO 0);
           clk :in std_logic;
           rst :in std_logic;
           output : out STD_LOGIC_VECTOR(15 downto 0));
end flc;

architecture structural of flc is
component tempfuzz is
port( te: in std_logic_VECTOR(7 DOWNTO 0);
      clk: in std_logic;
      rst :in std_logic;
      s1,s2,s3,s4,s5 :out std_logic_vector(15 downto
0));
end component;
component trfuzz is
port (tr: in std_logic_VECTOR(7 DOWNTO 0);
clk :in std_logic;
rst :in std_logic;
s6,s7,s8 :  out std_logic_vector(15 downto 0));
```

```
end component;
component fuzzyrulematrix is
port (s1,s2,s3,s4,s5,s6,s7,s8: in std_logic_vector(15
downto 0);
o1,o2,o3,o4,o5 : out std_logic_vector(15 downto 0));
end component;
component defuzzifier is
port (o1,o2,o3,o4,o5 :in std_logic_vector( 15 downto
0);
rst :in std_logic;
output : out std_logic_vector(15 downto 0));
end component;

signal cold,cool,mild,warm,hot :std_logic_vector(15
downto 0);
signal slow, moderate,fast :std_logic_vector(15 downto
0);
signal verylow,low,medium,high,veryhigh:
std_logic_vector( 15 downto 0);
begin
f1 :tempfuzz port map(te=>te,clk=>clk,rst=>rst,
s1=>cold,s2=>cool, s3=>mild, s4=>warm, s5=>hot);
f2 :trfuzz port map(tr=>tr,
clk=>clk,rst=>rst,s6=>slow,s7=>moderate, s8=>fast);
f3 :fuzzyrulematrix port
map(s8=>fast,s7=>moderate,s6=>slow,s5=>hot,s4=>warm,s3=
>mild,s2=>cool,s1=>cold,o5=>veryhigh,o4=>high,o3=>mediu
m,o2=>low,o1=>verylow);
f4 :defuzzifier port map
(o1=>verylow,o2=>low,o3=>medium,o4=>high,o5=>veryhigh,r
st=>rst,output=>output);
```

```
end structural;
```

SIMULATION RESULT OF FUZZY LOGIC CONTROLLER



CONCLUSION:

Temperature controller using standard fuzzy logic for temperature range of 0-125 degrees input(8 bit) has been designed. The unit has been coded in VHDL and simulated in XILINX Tool.