VHDL Implementation for Fuzzy Logic Temperature Controller

Neha Bahal
M-Tech student, Indira Gandhi Technical University for Women
Kashmiri Gate, Delhi

Shamim Akhter
Assistant Professor, Jaypee Institute of Information Technology
Noida, Uttar Pradesh

ABSTRACT
This paper, illustrates an implementation of fuzzy logic based temperature controller using VHDL in Modelsim 6.4. .Nowadays 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 humidity), the output (speed of air conditioner) is controlled based on relation between the fuzzy set (low, med, high, dry, moist, wet) of the 2 input variables.

Keywords
VHDL, temperature fuzzifier, humidity fuzzifier, fuzzy logic temperature controller.

1. INTRODUCTION
The concept of Fuzzy Logic (FL) was conceived at the beginning of the 70's by Lotfi Zadeh, a professor at the University of California at Berkley as a way of processing data by allowing the partial set membership rather than crisp set membership [7]. The word "fuzzy" is defined as "indistinct, imprecisely defined". Fuzzy systems are systems to be precisely defined. The real world is too complicated for precise descriptions therefore approximation (or fuzziness) must be introduced in order to obtain a reasonable accuracy. Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. This mapping provides a basis from which decisions can be made. 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)[6].

A Fuzzy Logic Controller (FLC) is an automatic controller that controls an object with desired behavior of inputs. The high growth of fuzzy logic applications led to the need of finding efficient way to hardware implementation [3].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.

1.1 Why Fuzzy Systems?
The fuzzy system improves the relative performance of a temperature control process with respect to conventional scheme [1]. The control action of FLC is defined in terms of if-then rules. These set of rules describes the system behavior. The fuzzy control systems are based on the theory of fuzzy logics and fuzzy sets [2], fuzzy set theory deals with non probabilistic uncertain issues. Several unique features of Fuzzy Logic

1. The output control is a smooth control function despite a wide range of input variations. So the system is inherently robust.
2. Any number of inputs can be processed and numerous outputs can be generated because of rule base operation.
3. Fuzzy Logic reduces the design development cycle.
4. Fuzzy Logic simplifies design complexity [6].
5. Fuzzy logic keeps the overall system cost and complexity low as it is not limited to a few feedback inputs and one or two control outputs can be programmed.
6. The fuzzy logic based systems and algorithms are widely employed in industrial control systems and engineering applications. The implementation of knowledge based control systems using human experiences is more practical than other methods [4], [5].

The purpose of this paper is to intelligently control the speed of an air controller system using fuzzy concepts. The system will be temperature and humidity dependent and will be able to control the output speed.

2. SYSTEM DESIGN
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.

Basic design of the model used in this work is given in the figure 1 below.

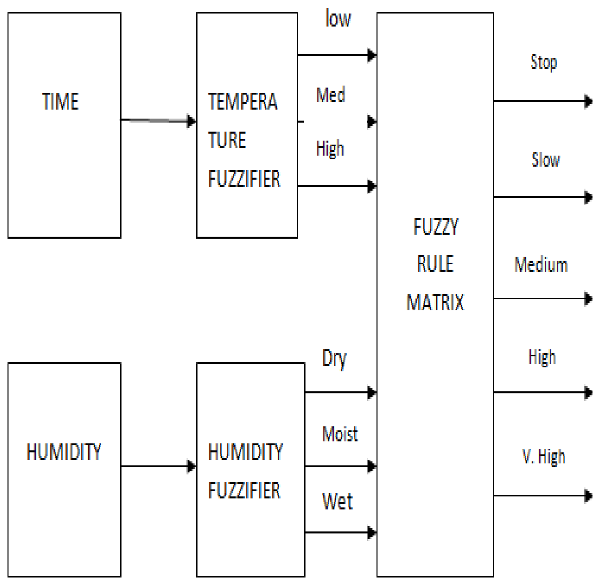

Fig. 1.Structural view of the system

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 low, med and high and 'Humidity fuzzifier' with membership functions dry, moist and wet.

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 stop, slow, medium, high and very high.

2.1 Membership Function

A membership function is a curve that defines how each point is mapped to a membership value in the input space (or degree of membership function) between 0 and 1. The input space is referred to as Universe of discourse. So the membership function involves:

1. Fuzzy sets which describes vague concepts.
2. Possibility of partial membership in fuzzy sets.
3. The degree of object belongs to a fuzzy set is denoted by membership value between 0 and 1.

Theoretical analysis of this problem indicates that while dealing with temperature controlling, operators can specify the temperature and humidity are the parameters on which decision has to be taken. Temperature and Humidity are the two input decision making variables for FIS (Fuzzy Interface System) [7, 9]. These are fuzzified into three sets respectively.

The temperature variable is fuzzified into three fuzzy sets (low, med and high) with a sample space of 0^0C to 50^0C. Similarly humidity variable is fuzzizfied into three fuzzy sets (dry, moist and wet) with humidity initial and final values (0 to 80). Membership functions can be taken in triangular and trapezoidal forms, due to large span over the UOD (Universe Of Discourse) between fuzzy sets of temperature; these are taken to be trapezoidal shapes.

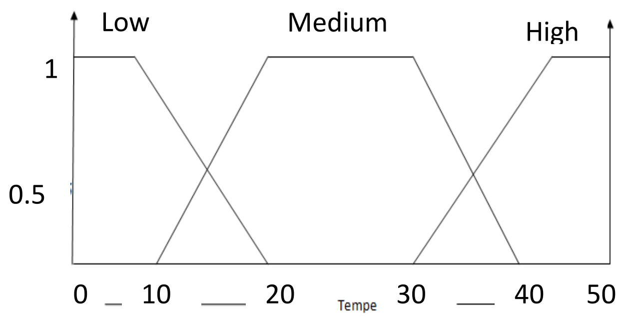

Fig.2. Fuzzily specified *TEMPERATURE*

Humidity is fuzzily specified in three triangular fuzzy sets namely, dry, moist and wet.

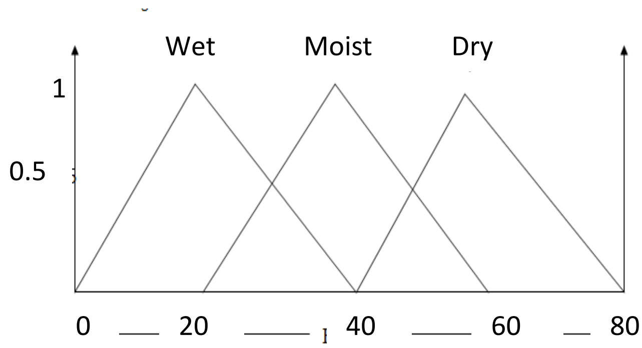

Fig.3. Fuzzily specified HUMIDITY

2.2 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 that compromise fuzzy logic. Fuzzy sets and operators are the subject and verbs of fuzzy logic.

Fuzzy sets provide an ultimate mechanism of communication between humans and computing environment [8].

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.

Fuzzy if-then rules for the proposed model are:

Table 1. Table Illustration of Fuzzy Rules

Inputs		Output
Temperature	Humidity	Speed of air conditioner
Low	Wet	Slow
Low	Moist	Stop
Low	Dry	Stop
Med	Wet	High
Med	Moist	Medium
Med	Dry	Slow
High	Wet	V. High
High	Moist	V. High
High	Dry	High

The rule base accepts two crisp input values, distributes the universe of discourse into region with each region containing two fuzzy variables, fires the rules and gives the output values corresponding to each output variable.

3. VHDL DESCRIPTION OF FUZZY LOGIC CONTROLLER

We have used VHDL language for designing behavioral architecture of fuzzy logic controller as:

1. It reduces the design time.
2. Different design choices are quickly explored.
3. Edit-compile-debug cycle is fastened [9].
4. It provides no insight into the physical implementation of entity but provides very flexible code that can be modified quickly [10].

The temperature fuzzifier, humidity fuzzifier and rule base used in our Fuzzy logic controller have been described with hand-optimized VHDL. Inputs to each variables containing membership functions with degrees for all input fuzzy sets.

ın the architecture of VHDL program membership degrees of input variables are defined for each fuzzy set. These values are included in the database of our program in binary format ranges as represented by fig 2 and fig 3 for each membership function i.e. temperature fuzzy sets are defined with the: low (0-20), med (10-40) and high (30-50). Similarly humidity fuzzy sets are defined in the ranges: wet (0-40) moist (20-60), dry (40-80. These values are taken from theoretical analysis.

4. SIMULATIONS

Modelsim 6.4 is used for design verification and logic synthesis. The simulation results obtained for the fuzzy logic temperature controller are shown below.

Fig. 4, Fig. 5, and Fig. 6 shows the results of the fuzzy logic temperature controller with two inputs, temperature fuzzifier, humidity fuzzifier and one output, rule base consisting of 9 rules for interfacing the two inputs as shown in the table1.

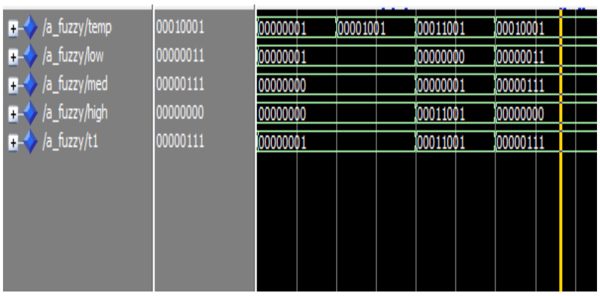

Fig 4: Simulation of Temperature Fuzzifier

Fig 4 shows the simulation result for temperature fuzzifier. The temp fuzzifier is loaded with different values of temperature as input from 0 to 50 represented using binary form in the database of VHDL code. For a particular value of temperature, based on the membership functions different values will be assigned to the three fuzzy sets of temperature (low, med, high) and the value of the fuzzy set having maximum degree will be assigned to T1 as temperature fuzzifier output.

As shown in the fig. the current value of Temp input is taken as '00010001' and obtained Fuzzy output is Med '00000111' because at present med has the highest degree among the three.

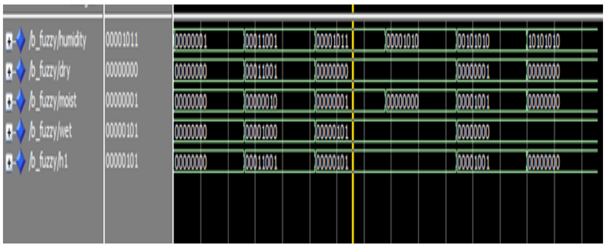

Fig 5: Simulation of Humidity Fuzzifier

Fig 5 shows the simulation result for humidity fuzzifier. The humidity fuzzifier is loaded with different values of humidity as input from 0 to 80 represented using binary form in the database of VHDL code. For a particular value of humidity, based on the membership functions different values will be assigned to the three fuzzy sets of humidity (dry, moist, wet) and the value of the fuzzy set having maximum degree will be assigned to H1 as humidity fuzzifier output.

As shown in the fig. the current value of Humidity input is taken as '00001011' and obtained Fuzzy output is Wet '00000101' because at present med has the highest degree among the three.

4.1 Overall system evaluation

System evaluation can be done by making fuzzy if-then rules with respect to the membership functions of input fuzzifier. Here output is the speed of temperature controller having five membership functions stop, slow, med, high, and very high. Only one of these functions is active with a particular value of the two fuzzifiers (Temperature and Humidity). VHDL rules are coded according to the illustrated rule matrix as shown in the table 1 and the simulation results are as follows:

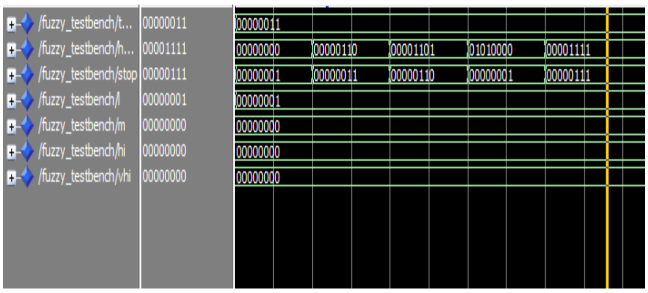

Fig 6: Simulation of Rule Matrix

Fig.6 shows that T1 has the temperature value = Low (00000011) and H1 has the humidity value = Moist (00001111) hence the outputs of the rule base i.e. speed of temperature controller is Stop with the reference from the rule base matrix shown in table 1. Similarly for other inputs of temperature and humidity the required speed of controller can be obtained.

5. CONCLUSION

This research work shows a modeling approach for describing fuzzy controller using VHDL simulation in Modelsim 6.4. The considered application in this work is a fuzzy logic temperature controller. It describes designing of a fuzzy controller using the fuzzifier and rule base parts of the system. The designed model controls the speed of the air conditioner successfully based on inputs given to temperature and humidity. This technique provides simplicity of adopting the codes for controller.

In future this work can be extended to real time applications. Further, more number of fuzzy sets for temperature and humidity can be included which will improve the accuracy of the controller. It can also be implemented for more number of input variables.

6. REFERENCES

[1] Ayala, I.L., and Solis, I.J. 1991. IEEE Transactions on Industry Applications. Volume: 27 , Issue: 1, pp:108 – 111

[2] H. J. Zimmermann, “Fuzzy Set and Its Applications, 3rd ed.” Kluwer Academic Publishers, Norwell, MA. (1996).

[3] MS Anand, “Design and Implementation of Fuzzy Controller on FPGA”, http://www.mecs-press.org/ijisa/ijisa-v4-n10/v4n10-4.html.

[4] A. Barriga, S. Sanchez-Solano, P. Broxl, A. Cabrera, and I. Baturone1, “VHDL High Level Modeling and Implementation of Fuzzy Systems", International workshop on fuzzy logic and applications, vol. 2955, pp. 11-18, 2003.

[5] Thomas Hollstein, Saman K. Halgamuge and Manfred Glesner, “Computer Aided Design of Fuzzy Systems based on generic VHDL Specications”, IEEE Transactions on Fuzzy Systems, vol. 4, no. 4, pp. 403-417, 1996.

[6] A course in fuzzy system and control; Book by Li-Xin Wang.

[7] Antonio Di Stefano and Costantino Giaconia “An FPGA-Based Adaptive Fuzzy Coprocessor”, Computational intelligence and bioinspired systems, vol. 3512, pp. 590-597, 2005.

[8] S. Gottwald, “Mathematical fuzzy logic as a tool for the treatment of vague information” Inf. Sci. 172:1-2 (2005) 41-71

[9] Brown S. & Vranesic Z.; Fundamentals of Digital Logic with VHDL Design; MCGrow-Hill Company, 2002.

[10] Thararam S. Muk; Design and FPGA Implementation of an Adaptive Demodulator; M.SC. Thesis, University of Kansas; 2000.