Design and Implementation of an Embedded System for Fuzzy Logic based Traffic Light Controller

Prakash Tonse
U.G Student
Dept. of E&CE
S.D.M.C.E.T, Dharwad-02
Karnataka, India

Shubhalakshmi Shetty
U.G Student
Dept. of E&CE
S.D.M.C.E.T, Dharwad-02
Karnataka, India

Shruti S.R.
U.G Student
Dept. of E&CE
S.D.M.C.E.T, Dharwad-02
Karnataka, India

Siddalingesh S. Navalgund
Assistant Professor
Dept. of E&CE
S.D.M.C.E.T, Dharwad-02
Karnataka, India

ABSTRACT
Traffic congestion has become a common scenario in today's day to day experience and this can be owed to the fact that most of the junctions using conventional crisp Traffic light controllers. Fuzzy Logic the basis of Soft computing can be considered to be the superset of the Crisp Logic which takes into account the whole spectrum of possibilities. Fuzzy Logic best handles the uncertainty in the inputs and can give best possible outcomes compared to Aristotelian logic. It also has an upper hand as it gives the system a human like thinking. If Fuzzy logic can be applied to conventional traffic controlling one can see the dynamic control of the same. Furthermore fuzzy logic makes the system adaptable and intelligent as it can best handle the irregularities in the input. This paper focuses on the design and implementation of an Embedded System for Fuzzy Logic based Traffic Light Controller (TLC) using VHDL which includes the implementation of Fuzzifier, Fuzzy Inference Engine and the Defuzzifier on FPGA using VHDL. The simulations have been carried out using ModelSim 6.1 simulator and synthesis has been carried out using Xilinx XST. The implementation is done on Spartan xc3s400-5pq208 FPGA and a maximum Frequency of 487.936MHz is obtained.

Keywords
defuzzification, Fuzzy logic, FPGA, fuzzification, Soft Computing, Traffic Light Controller**.**

1. INTRODUCTION
Fuzzy logic refers to a logical system that generalizes the classical two-value logic for reasoning under uncertainty. It is a system of computing and approximate reasoning based on a collection of theories and technologies that employ fuzzy sets, which are classes of objects without sharp boundaries. More specifically, fuzzy logic generalizes the crisp true-and- false (or black-and-white) concept fundamental to classical logic to a matter of degree. The two key features of fuzzy logic include: (a) A mathematical formalism for representing human knowledge involving vague concepts, and (b) a natural but effective mechanism for systematically formulating cost-effective solutions to complex problems characterized by uncertainty or imprecise information

A natural question is why fuzzy logic is effective. The answer is simple and has two aspects, both of which stem from the two features described above. The first aspect is that fuzzy logic is well suited for controlling a process or system that is too nonlinear or too poorly understood to use conventional control designs. The second aspect is that fuzzy logic enables control engineers to systematically implement control strategies used by human operators with experience and expertise.

The membership value is indicated as μ, and is mapped into the range from 0 to 1 as shown in equation (1).[1]

$\mu_A(x) \in [0,1]$……..…….(1)

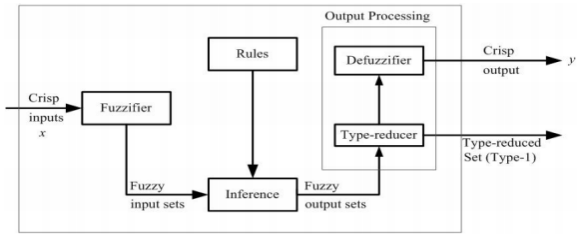

Fig 1: Block Diagram of a Fuzzy Logic Controller

The general block diagram of a fuzzy logic controller is shown in fig. 1. It consists of a normaliser, fuzzifier block along with knowledge base, inference engine block along with a rule base, defuzzifier block. A fuzzifier is a mapping from a real-valued point $x^* \in U$ to a fuzzy set A' in U. Many fuzzifiers are defined, some of which are singleton fuzzifier, Gaussian fuzzifier, triangular fuzzifier etc. The Gaussian and triangular fuzzifiers can suppress noise in the input.

A defuzzifier is defined as a mapping from fuzzy set B' in V (which is the output of the fuzzy inference engine) to crisp point $y^* \in V$. Many defuzzifiers are available, some of which are CG method, center of sums, mean-max membership method, weighted average method etc. Plausibility, computational simplicity and continuity are the three criteria considered while choosing a defuzzifier. The aggregation of

rules for finding the fuzzy output may be written as in equation (2).[2]

$\mu(y) = V \{ \mu_A(y) \wedge \mu_B(y) \}$ ……..……..(2)

The fuzzy inference engine is the heart of the controller. It consists of the rule-base. The rules are constructed using fuzzy IF-THEN-ELSE constructs. In the literature, conditional, unconditional and simple assignment types of rules have been reported[1].

The rules can have conjunctive or disjunctive set of premises. Basically two types of inference engines exist; Mamdani type and Sugeno type. These two types of inference systems vary somewhat in the way outputs are determined.

, which finds the centroid of a two-dimensional Mamdani's fuzzy inference method is the most commonly seen fuzzy methodology[2]. In Mamdani inference engine, after the aggregation process, there is a fuzzy set for each output variable that needs defuzzification. It's possible, and in many cases much more efficient, to use a single spike as the output membership functions rather than a distributed fuzzy set. This is sometimes known as a *singleton* output membership function, and it can be thought of as a pre-defuzzified fuzzy set. It enhances the efficiency of the defuzzification process because it greatly simplifies the computation required by the more general Mamdani methodfunction. Rather than integrating across the two-dimensional function to find the centroid, we use the weighted average of a few data points. Sugeno-type systems support this type of model. In general, Sugeno-type systems can be used to model any inference system in which the output membership functions are either linear or constant [3,4].

2. LITERATURE SURVEY

On the exhaustive survey of literature it was found that varied number of approaches has been carried out to define the Fuzzy systems to control various applications. [5] Concerns an implementation of a fuzzy logic controller (FLC) on a reconfigurable field-programmable gate array (FPGA) system. In the proposed implementation method, the FLC is partitioned into many temporally independent functional modules, and each module is implemented individually on the FLC automatic design and implementation system, which is an integrated development environment for performing many subtasks such as automatic VHSIC hardware description language description, FPGA synthesis, optimization, placement and routing, and downloading. [6] A software for synthesizing fuzzy controllers into Boolean equations is developed. The Boolean equations are then realized on the FPGA by programming it. The speed of the fuzzy controller is determined by the response time of the FPGA circuit that realizes the Boolean equations. A speed of 50M FLIPS is achieved. In [7] the basic components of the fuzzy logic controller are designed using VHDL and a Xilinx virtex FPGA is used for implementation. The fuzzy Iogic controller with an 8-bit input, 8-bit output is tested by controlling single disk of an ECP torsional plant. The paper in [8] presents different approaches to design and implement a Fuzzy Logic Controller (FLC) for an intelligent parking system (IPS). With the software approach, FLC is performed on a flexible FPGA soft core processor. [9] proposes a novel method of implementing fuzzy logic operations in digital hardware using bit-serial arithmetic.

3. FUZZIFIERS AND DEFUZZIFIERS

Various types of Fuzzifiers and Defuzzifiers are described in the literature[1] out of which the ones chosen for implementation on FPGA are described below. Fuzzifiers form the basis for converting the crisp values into fuzzy quantities and form the initial entity in any soft computing applications. Here the crisp values are suitably mapped onto the fuzzy spectrum.

3.1 Types of Fuzzifiers

Fuzzifiers can be of varied types. Usually a look up table approach based on intuitiveness is applied where in the membership values are assigned based on experience. Even then there are fuzzifiers like Triangular, Gaussian, Rectangular, Trapazoidal fuzzifiers which can be applied it the fuzzy quantities are varying in a defined manner [1]. Some of the fuzzifiers used for implementation are explained below.

a) Triangular Fuzzifier: This type of fuzzifier has a triangular variation as can be seen the Fig. 2.

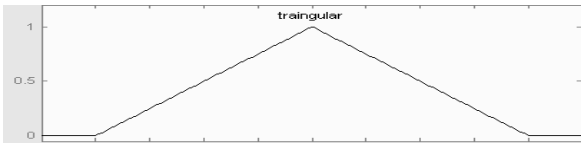

Fig 2: Triangular Fuzzifier

b) Trapezoidal Fuzzifier: This type of fuzzifier is having a characteristic variation of trapezoid as shown in Fig. 3

Fig 3: Trapezoidal Fuzzifier

3.2 Types of Defuzzifiers

If Fuzzification gives us the fuzzy value to handle the uncertainities then the Defuzzifier gives us the crisp value which can be understood by any processor or device in real world and hence forms one of the very important blocks of any Fuzzy logic based inference systems used in Soft Computing[2]. There are varieties of Defuzzifiers available like min-max method, center of sums methods etc., and the one chosen for implementation is described below.

Centroid method: This procedure (also called center of area, center of gravity) is the most prevalent and physically appealing of all the defuzzification methods [1]; it is given by the algebraic expression.

$y^* = (\int y \ \mu_B(y)\, dy) \div (\int \mu_B(y)\, dy)$……...(4)

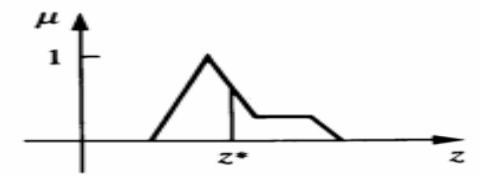

Fig 4: Centroid Defuzzification Method

4. VHDL IMPLEMENTATION

4.1 Entity Declaration

In this section, the implementation of Traffic Light Controller (TLC) using VHDL is described. In this prototype implementation, the TLC receives inputs in the form of density at respective lanes as shown in the Fig. 5. The outputs will be in the form of time delay for which the green light should be on at the respective lanes with corresponding densities. A reset pin (rst) is provided to reset the controller at any point in time.

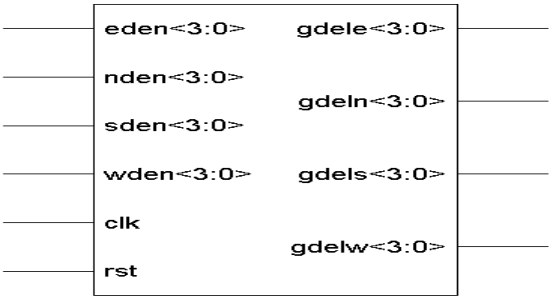

Fig.5: Entity for Traffic Light Controller.

4.2 *Programming the TLC*

The programming environment for TLC is based on VHDL. The behavioral description is analyzed simulated and synthesized onto Xilinx FPGAs. An application for the prototype implementation is developed by writing behavioral description in VHDL and the description is iteratively refined and debugged with the simulator available i.e., ModelSim version 6.1. After the description code is verified to be functionally correct by simulation, it is translated onto a Xilinx net list form. The net list is then mapped onto FPGA architecture by automatic partition, placement and routing tool to form a loadable FPGA object module[10]. A static timing analysis tool is then applied to the object module to determine maximum operating speed for all the implementations.

5. RESULTS AND DISCUSSIONS

5.1 Simulation Results

The behavioral description of the Fuzzifiers, Defuzzifier and the entire TLC are written and synthesized using Xilinx XST version 6 and simulated and functionally verified on the ModelSim simulator version SE 6.1. The simulation results of the Fuzzifiers is shown in Fig. 6. Similarly the defuzzifier can be seen in Fig. 7. The Simulation results for the entire Fuzzy Logic based Traffic Light controller can be seen in Fig 10. This includes the entire fuzzification, rule base and the defuzzification part. The proposed TLC is provided with four inputs corresponding to densities at North, South, East and West lanes. This TLC is also provided with a reset signal so that the TLC can be reset as and when required. The controller works on the event of clock (clk). The simulation results obtained are up to the expectations and hence the functionality of the code is verified.

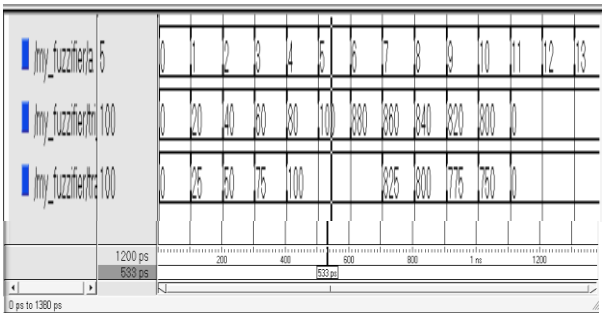

Fig.6: Simulation Results for Fuzzifiers.

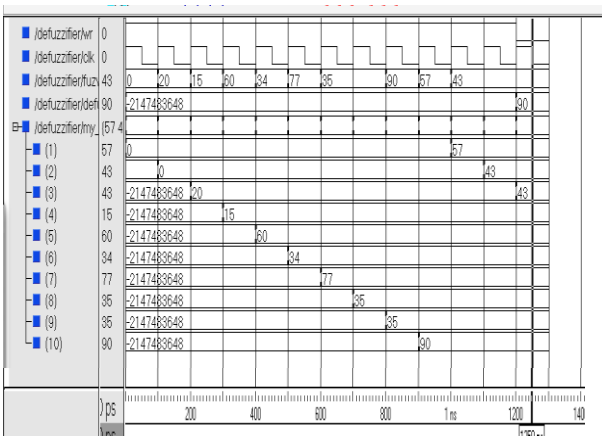

Fig.7: Simulation results for Defuzzification

5.2 . IMPLEMENTATION ON FPGA

As the entire code is synthesizable, the synthesis is carried out on the FPGA (Spartan **3s400pq208-5**). The entire design is optimized for speed and area using Xilinx device family Spartan. The device utilization, timing report, logic distribution and other details are given in table 1. The implementation of the same on Spartan 3 kit can be seen in fig 8.

TABLE 1.
DEVICE UTILIZATION SUMMERY FOR TLC

Logic Utilization and Distribution	Used	Available	%Utilization
Number of Slices	681	3584	19%
Number of Slice Flip Flops	4	7168	0%
Number of 4 input LUTs	1073	7168	14%
Number of bonded IOBs	10	141	7%
Number of MULT 18X18s	1	16	12%
Minimum period: 2.049ns (Maximum Frequency: 487.936MHz)			
Minimum input arrival time before clock: 86.133ns			
Maximum output required time after clock: 6.280ns			

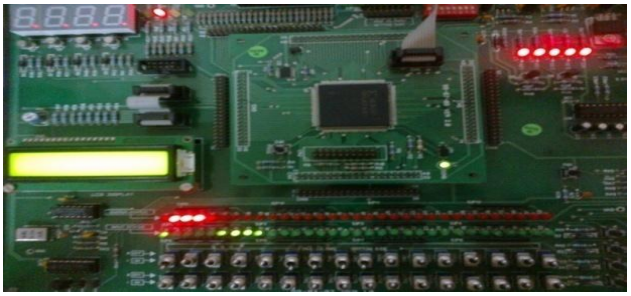

Fig 8. FPGA implementation of TLC in Laboratory.

5.3 Technology and RTL Schematic

Before implementing the code on FPGA, the schematics were generated for Fuzzifiers and Defuzzifiers. The RTL schematic for Fuzzifiers can be seen in Fig. 12. The Register Transfer Logic appears simple showing the efficient utilization of resources.

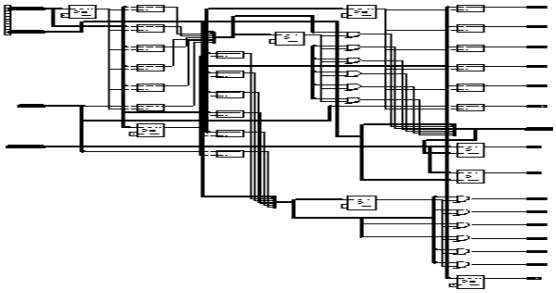

Fig. 9. RTL Schematic for Traffic Light Controller.

5.4 Result Analysis

The proposed traffic light controller works on the principle of round robin fashion. That is to say the traffic lights will be switched either in clockwise for anticlockwise direction as may be the requirement. If any change in the traffic density occurs, the value is stored in the register and accordingly the wait time for the next lane is calculated by the FLC. The existing traffic light controllers are crisp in nature and use the crisp delay values for each lane thus making it inefficient. Also if one looks at the device utilization summary of the proposed designed, it can be seen that optimum utilization of the resources has been achieved. The maximum working frequency being 487.936MHz gives the user with high performance and precise computing of the delay values with accurate switching.

6. CONCLUSION

In this paper, an attempt has been made to design and implement and embedded system for Fuzzy Logic based Traffic Light Controller on FPGA using VHDL. Since Fuzzy logic best captures the ambiguity in the input when implemented in VLSI domain can give us and intelligent embedded system which can perform in more efficient way. This work is confined to writing behavioral description for Fuzzifiers, Fuzzy Inference Engine and Defuzzifiers for Traffic Light Control Application and making the code synthesizable. The synthesis results obtained show satisfactory resource utilization for the prototype implemented. This can be further converted to ASICs and can be used as a part of the system in any of the required traffic light control scenario.

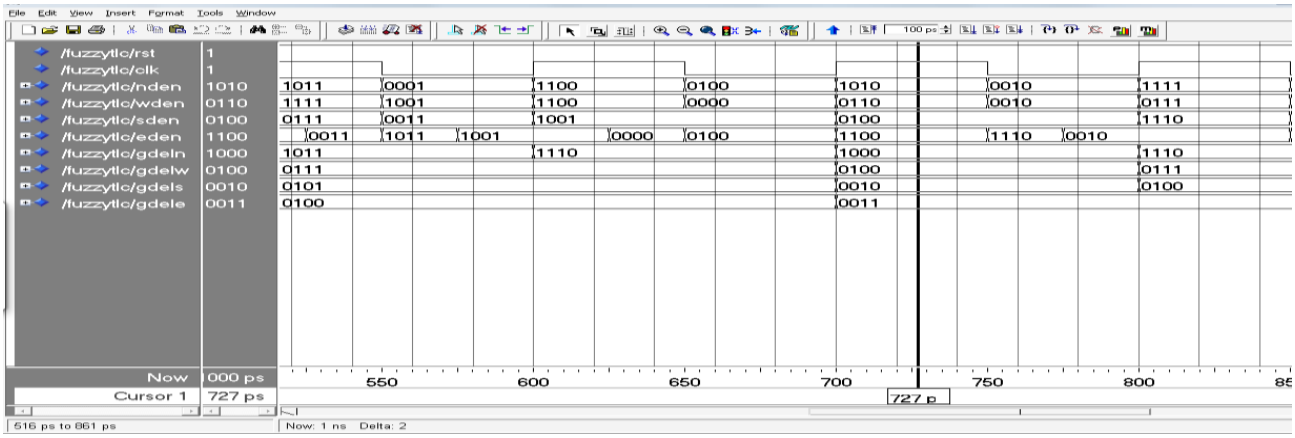

Fig.10: Simulation results for proposed Fuzzy logic based Traffic Light controller.

7. ACKNOWLEDGMENTS

The authors thank the authorities of Sri Dharmasthala Manjunatheshwara College of Engineering and Technology, Dhavalagiri, Dharwad, Karnataka, India for encouraging him for this research work. The author especially thanks Prof. S. S. Navalgund for teaching the course on Fuzzy Logic and VHDL and also for motivation and timely guidance.

8. REFERENCES

[1] Timothy Ross, "Fuzzy Logic with Engineering Applications", McGraw Hill Publications,

[2] Shubhalakshmi Shetty, Shruti S R., Prakash R. Tonse, Siddalingesh. S. Navalgund,"*A Comparative analysis of Mamdani and Sugeno Fuzzy Rule Based Washing Machine Controller*". Proceedings of ICETE-11, pp:440-445.

[3] Li Xin Wang, *"A Course in Fuzzy Systems and Control", Prentice Hall International.*

[4] Wu Wei and Yi Zhang *," FL-FN Based Traffic Signal Control",* 0-7803-7280-8/02/$10.00, 2002 IEEE Transaction.

[5] Daijin Kim, *Member, IEEE* , An Implementation of Fuzzy Logic Controller on the Reconfigurable FPGA System , *IEEE TRANSACTIONS ON INDUSTRIAL*

ELECTRONICS, VOL. 47, NO. 3, JUNE 2000 pp:703-715.

[6] Mahmoud A. Manzoul and Dinesh Jayabharathi , FUZZY CONTROLLER ON FPGA CHIP, 62901 0-7803-0236-2 192 33.00 *0* 1992 IEEE pp:1309-1316.

[7] Sameep Singh and Kuldip *S.* Rattan , Implementation of a Fuzzy Logic Controller on an FPGA using VHDL 0-7803-7918-7/03/$17.00 @ 2003 IEEE pp:110-115..

[8] Mohamed Slim Masmoudi, Insop Song, Fakhreddine Karray, Mohamed Masmoudi, Nabil Derbel, HARDWARE/ SOFTWARE APPROACHE FOR THE FPGA IMPLEMENTATION OF A FUZZY LOGIC CONTROLLER 0-7803-9727-4/06 2006 IEEE.

[9] Scott Dick, Vincent Gaudet, Huiqing Bai, Bit-Serial Arithmetic: A Novel Approach to Fuzzy Hardware Implementation, 978-1-4244-2352-1/08/$25.00 ©2008 IEEE

[10] Douglas Perry, VHDL Programming by Example, Tata McGrawHill Publishers, VI edition

AUTHOR'S PROFILE

Prakash Tonse is pursuing his 4th year of undergraduate studies in the Dept. of E&CE at SDMCET, Dharwad. His research interests include VLSI, Digital Circuits and Fuzzy Logic Applications. Mr. Prakash is currently the student member of Institution of Electronics and Telecommunication Engineers (IETE).

Shruti S.R. is pursuing her 4th year undergraduate studies in the Dept. of E&CE at SDMCET, Dharwad. Her research interests include VLSI and Fuzzy Logic Applications.

Shubhalakshmi Shetty is pursuing her 4th year of undergraduate studies in the Dept. of E&CE at SDMCET, Dharwad. Her research interests include VLSI and digital electronics.

Siddalingesh S. Navalgund received Bachelor of Engineering in Electronics and Communication Engineering from Karnataka University Dharwad in 2000 and Master of Technology in Microelectronics and Control Systems from Visvesvaraya Technological University, Belgaum in 2004

After spending two years in the industry, he joined as a faculty in N. M. A. M. in Institute of Technology, Nitte, Karnataka. Since 2005, he is serving as faculty in the Dept. of E&CE, S. D. M. College of Engineering and Technology, Dharwad, Karnataka. His main research interests include VLSI design, embedded & fuzzy systems and digital signal processing.
Mr. Navalgund is currently the life member of Institution of Electronics and Telecommunication Engineers (IETE), Indian Society for Technical education (ISTE) and Institution of Engineers India (IE-I).
2004