International Journal of Engineering Sciences & Emerging Technologies, June 2012.
ISSN: 2231 – 6604 Volume 2, Issue 2, pp: 53-58 ©IJESET

SIMULATION OF 64-BIT FUZZY INFERENCE PROCESSOR USING VHDL FOR ATM

Kamini M. Sayam[1], S. V. Bhalerao[2]
[1]Department of Electronics Engineering, RTM University, Nagpur City, India

ABSTRACT

The Fuzzy inference systems are becoming widely used in the field of real time control in various application, thanks to the potential offered by fuzzy logic and the possibility of implementation using very large scale integration (VLSI) technology, we deal a fuzzy inference system with VHDL code is designed to achieve traffic control or to correct the traffic in high speed Asynchronous Transfer Mode (ATM) network. The design of 64-bit fuzzy processors, such as fuzzifier ,rule engine and defuzzifier . so that to use this designed system in a real time application so that the speed of the process increases to a very far level due to which the overall efficiency must be increased so that it should be very beneficial for user. By designing 64-bit fuzzy inference processors we get the corrected traffic at output of ATM module. For the better result and to speed up the processing time for ATM module we implement the same system with VHDL.

KEYWORDS: *Fuzzifiers, Rule engine, Defuzzifiers, VHDL, ATM Network.*

I. INTRODUCTION

Today most fuzzy logic systems are implement software on microprocessors. If there is a need for higher operation speed, we can add a fuzzy processor (FP). For 8-bit and 16-bit systems the FP speedup the system. But when using a 32 bit or 64-bit microprocessor, an optimized fuzzy controller algorithm implement as software is faster than the simple FP. Our idea of a 64-bit fuzzy system was to minimize the needed hardware requirement for the fuzzy processor by using already existing components of the 64-bit microprocessors and implement only those components which speed-up the fuzzy algorithm. This project covers the circuit and architecture level designing of various components of the fuzzy processors, such as, fuzzifiers, rule engine and defuzzifiers, . A comparative analysis of the performance of these components will be performing. Further, it is seen that the design emphasis should be more on inference engine performance and defuzzification units, because of the complexity of computations handled by them. The optimization in these units results in a significant improvement in the overall performance of the system. The methods and approaches adapted for the realization of four important components of fuzzy processors, such as, fuzzifier, defuzzifier, inference engine and rule-base has been thoroughly studied. The overall performance of the processors has a significant dependence on the performance and optimization in these units. The architecture and circuit level designing of the fuzzy processors finally realized in the form of FPGAs using VHDL. Fuzzy logic is being increasing interest in various fields of application such as process control, decision-making support systems and According to the field of application, fuzzy systems feature time constraints that may differ considerably. The Key features of the designed project to get corrected traffic at the output of the ATM. The design is implemented using VHDL and verified on Xilinx ISE simulator.

1.1 Asynchronous Transfer Mode (ATM)

Asynchronous Transfer Mode (ATM) is a standard switching technique, designed to unify telecommunication and computer networks. It uses asynchronous time-division multiplexing, and it encodes data into small, fixed-sized cells. This differs from approaches such as the Internet Protocol or Ethernet that use variable sized packets or frames. ATM provides data link layer services that run over a wide range of OSI physical Layer links. ATM has functional similarity with both circuit switched networking and small packet switched networking. It was designed for a network that must handle both traditional high-throughput data traffic (e.g., file transfers), and real-time, low-latency content such as voice and video. ATM uses a connection-oriented model in which a virtual circuit must be established between two endpoints before the actual data exchange begins. ATM is a core protocol used over the SONET/SDH backbone of the public switched telephone network (PSTN) and Integrated Services Digital Network (ISDN), but its use is declining in favour of All IP.

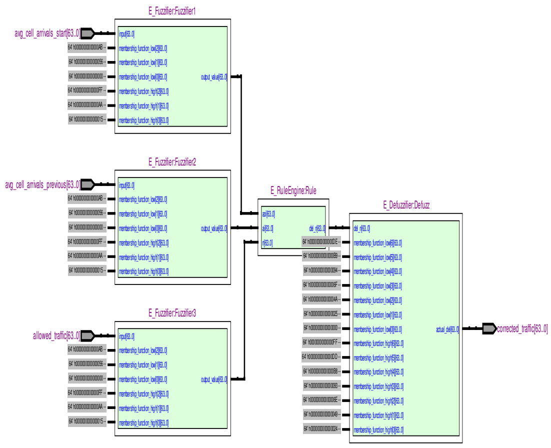

Fig. 1 The designed system

II. FUZZY CONTROL SYSTEM FOR ATM NETWORK

Fig. 1 shows the designed system which has basically three units along with ATM module. The ATM module consist of three fuzzifier, one rule engine and one defuzzifier

A . Fuzzification - Fuzzification is the process of decomposing a system input and/or output into one or more fuzzy sets, here we used three fuzzification unit , the first fuzzifier has an input of average cell arrived at the previous and had three membership function .the second fuzzifier has input as the average cell arrived at the starts of window and three membership function similarly at the third fuzzifier input is allowed traffic and again three membership function .the types of curves can be used, but triangular or trapezoidal shaped membership functions are the most common because they are easier to represent in embedded controllers. Each fuzzy set spans a region of input (or output) value graphed with the membership. Any particular input is interpreted from this fuzzy set and a degree of membership is interpreted. The membership functions should overlap to allow smooth mapping of the system. The process of fuzzification allows the system inputs and outputs to be expressed in linguistic terms so that rules can be applied in a simple manner to express a complex system

B .Rule Engine - The selection of rule base is based on our experience and beliefs on how the system should behave. Design of a rule base is two-fold: First, the linguistic rules are set afterwards, membership functions of the linguistic values are determined. The trade-off involving the design of the rule base is to have a set of minimum number of linguistic rules representing the control surface with sufficient accuracy to achieve an acceptable performance. Recently, in the fuzzy control literature, some formal techniques for obtaining a rule base by using Artificial Neural Networks or Genetic Algorithms have appeared. Nevertheless, we have used the conventional trial and error approach under the guidance of some design rules of thumb. Usually, to define the linguistic rules of a fuzzy variable, Gaussian, triangular or trapezoidal shaped membership functions are used. Since triangular and trapezoidal shaped functions offer more computational simplicity, we have selected them for our rule base. Then, the rule base is tuned by observing the progress of simulation

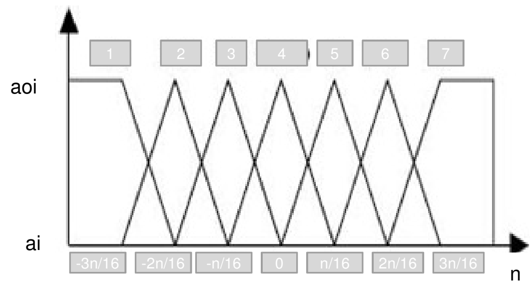

Fig 2 :Fuzzy policing model

The parameters used in fuzzy policing model are made up of linguistic variables and fuzzy sets, while control action is described by a set of fuzzy conditional rules. The input variables for the fuzzy policing model are:

1)aoi---the average number of cell arrivals at the start per window ;

2)ai--- the average number of cell arrivals in the previous

3)n--- the controller threshold (allowedt cells per window) in the previous window.

The output variable is:

Del_n---the variation to be made to the threshold i in the next window.

2.1Table 1: Rule Table -Table 1 shows the fuzzy conditional rules for fuzzy policer model. Rule 1 in Table 1 has to be interpreted as If (aoi is Low)and (n is High) and (ai is Low) Then (Del_n is Positive Big)

Table 1 Fuzzy Rule Base Here we consider for low value 0,for medium it is 1and for high it is 2 The output Del_n is according to the fuzzy policing model as shown in Fig 2

Sr.No	aoi	ai	*n*	*Del_n*
1	0	0	2	7
2	0	1	2	5
3	0	2	2	4
4	1	0	1	7
5	1	1	1	5
6	1	2	1	4
7	1	0	2	7
8	1	1	2	4
9	1	2	2	1
10	2	0	0	7
11	2	1	0	6
12	2	2	0	5
13	2	0	1	7
14	2	1	1	6
15	2	2	1	1
16	2	0	2	2
17	2	1	2	3
18	2	2	2	4

To understand how the fuzzy rules in the knowledge base reproduce the logic process an expert knowledge in the field would apply. Let us consider just one case: In case that the source is fully respectful (Aoi is low) which involves rules 1-3, Ni is necessarily high due to the fact that the source has gained credit. Thus, if the number of cells that arrived in the last window is low or medium, that is, the source continues nonviolating behavior, its credit is increased rules 1,2; vice versa, if Ai is high, a sign of a possible beginning of violation on the part of the source or an admissible short term statistical fluctuation, the threshold value remains unchanged rule 3. A MatLab program has been developed to simulate the behaviour of this fuzzy policing model. The results are discussed as follows. The x axis represents i-th window and y axis represents number of cells. From Figure 4 (end of the paper) we can see that the long term traffic Aoi and short term traffic Ai do not exceed contract value N so that maximum credit Ni is given by fuzzy logic based policer. In Figure 5 (end of the paper) we can see that the long term traffic Aoi and short term traffic Ai exceed the contract N so minimum credit Ni can be given by fuzzy logic based policer. We have shown that that this fuzzy logic based policer can well perform traffic control as described in 18 fuzzy rules in Table 1.

C. Defuzzification

Fuzzy logic is a rule-based system written in the form of horn clauses (*i.e.*, if-then rules). These rules are stored in the knowledge base of the system. The input to the fuzzy system is a scalar value that is fuzzified. The set of rules is applied to the fuzzified input. The output of each rule is fuzzy. These fuzzy outputs need to be converted into a scalar output quantity so that the nature of the action to be performed can be determined by the system. The process of converting the fuzzy output is called defuzzification. Before an output is defuzzified all the fuzzy outputs of the system are aggregated with an union operator. The union is the *max* of the set of given membership functions and can be expressed as

$$\mu_A = \bigcup_i (\mu_i(x)) \tag{2.1}$$

There are many defuzzification techniques but primarily only three of them are in common use There are many defuzzification techniques but primarily only three of them are in common use. These defuzzification techniques are discussed below in detail.

Maximum Defuzzification Technique

This method gives the output with the highest membership function. This defuzzification technique is very fast but is only accurate for peaked output. This technique is given by algebraic expression as

$$\mu_A(x^*) \geq \mu_A(x) \text{ for all } x \in X \tag{2.2}$$

where *x** is the defuzzified value. This is shown graphically in Figure 2

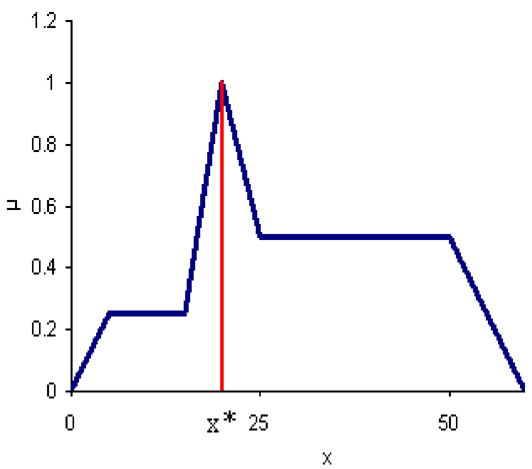

Figure 2 Max-membership defuzzification method

Centroid Defuzzification Technique

This method is also known as center of gravity or center of area defuzzification. This technique was developed by Sugeno in 1985. This is the most commonly used technique and is very accurate. The centroid defuzzification technique can be expressed as

$$x^* = \frac{\int \mu_i(x)\, x\, dx}{\int \mu_i(x)\, dx} \qquad (2.3)$$

where x^* is the defuzzified output, $\mu_i(x)$ is the aggregated membership function and x is the output variable. The only disadvantage of this method is that it is computationally difficult for complex membership functions. This method is illustrated in

Weighted Average Defuzzification Technique

In this method the output is obtained by the weighted average of the each output of the set of rules stored in the knowledge base of the system. The weighted average defuzzification technique can be expressed as

$$x^* = \frac{\sum_{i=1}^{n} m^i w_i}{\sum_{i=1}^{n} m^i} \qquad (2.4)$$

where x^* is the defuzzified output, m^i is the membership of the output of each rule, and w_i is the weight associated with each rule. This method is computationally faster and easier and gives fairly accurate result. This defuzzification technique is applied in fuzzy application of signal validation in and fuzzy application on power.

Defuzzification is the process of producing a quantifiable result in fuzzy logic, given fuzzy sets and corresponding membership degrees. It is typically needed in fuzzy control systems. These will have a number of rules that transform a number of variables into a fuzzy result, that is, the result is described in terms of membership in fuzzy sets

After fuzzy reasoning we have a linguistic output variable which needs to be translated into a crisp value. The objective is to derive a single crisp numeric value that best represents the inferred fuzzy values of the linguistic output variable. Defuzzification is such inverse transformation which maps the output from the fuzzy domain back into the crisp domain. Some defuzzification methods tend to produce an integral output considering all the elements of the resulting fuzzy set with the corresponding weights. Here we used Centroid methods take into account just the elements corresponding to the maximum points of the resulting membership functions.

III. CONCLUSIONS

This project is basically designed and implement the architecture of 64-bit fuzzy inference processor for ATM n/w using VHDL so that to increase the computational speed of the processor highly reliable and we get the corrected traffic at the output ,in this way we control the traffic of the ATM network with efficiency and low complexity controllers , power consumption & processing throughput

Simulation result

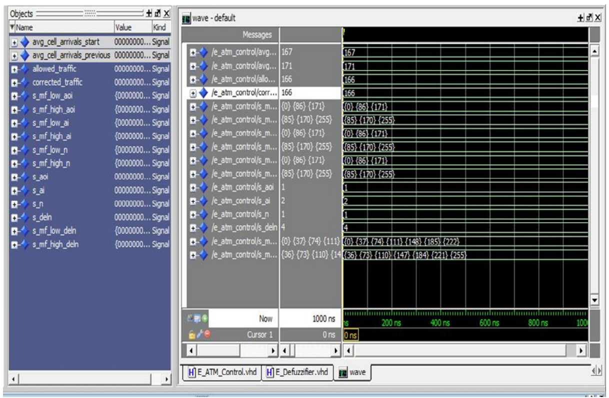

Fig .3. Simulation result of 64 bit fuzzy inference processor for ATM network

REFERENCES

[1] H. Ikeda, Y. Hiramoto and N. Kisu Electronics Research Laboratory, Central Engineering Laboratories, Nissan Motor Co., Ltd. 1 Natsushima-Cho. Yokosuka 237, a Fuzz Y Inference Processor R with an "Active - Rule-Driven" Architecture.

[2] J. Fattaruso, S. S. Mahant-Shetti, J. B. Barton A Fuzzy Logic Inference Processor, Semiconductor Process And Design Center Texas Instruments Inc. Pob 655474 Ms 446 Dallas, Tx 75265

[3] Herbert Eichfeld*, Martin Klimke', Manfred Menke*, Jurgen Nolles" And Thomas Kunemund*, A General-Purpose Fuzzy Inference Processor, *Siemens Ag, Corporate Research and Development, D-8 1730 Munich, Germany "Siemens Ag, Semiconductor Group, D-81617 Munich, Germany.

[4] J.S. Ho, M.H. Liiii And K.T. Lau School Of Electrical And Electronic Engineering Nanyang Teclnological University, Singapore ,Simplified Architecture Of A Fuzzy Inference Processor 0-81864260-2/9$30 3.00 0 1993 IEEE

[5] Ansgar P. Ungering, Karl Goser University Of Dortmund, Ls-Be, Emil-Figge-Str. 68,44227 Dortmund, Germany , Architecture Of A 64-Bit Fuzzy Inference Processor, , 0-7803-1896-X/94 $4.00 01994 IEEE

[6] Herbert Eichfeld, Martin Klimke, Manfred Menke, Jurgen Nolles, Thomas Kunemund Siemens Ag A General-Purpose Fuzzy Inference Processor, 0272-1732/95/$04.00 8 1995 IEEE.

[7] Ctp Song, Sf Quigley, S Pammu, School Of Electronic And Electrical Engineering, University Of Birmingham, Birmingham, B 15 2tt, England: A Novel Cmos Analogue Fuzzy Inference Processor, 0-7803-4455-3/98/$10.00 0 1998 IEEE

[8] Masahiro Sakamoto, Daisuke Hamano And Mititada Morisue, Faculty Of Information Sciences Hiroshima City University, Hiroshima, 73 1-3 194 Japan, 1999 IEEE International Fuzzy Systems Conference Proceedings, August 22-25, 1999, Seoul, Korea, A Study Of A Radix-2 Signed-Digital Fuzzy Processor Using The Logic Oriented Neural Networks.

[9] Giuseppe Ascia And Vincenzo Catania, Istituto Di Informatica E Telecomunicazioni, V.Le A. Doria, 6-95125-Catania-Italy, 1999 IEEE International Fuzzy Systems Conference Proceedings August 22-2j, 1999, Seoul, Korea, A High Performance Processor For Applications Based On Fuzzy Logic.

[10] Giuseppe Ascia And Vincenzo Catania Universith Di Catania Istituto Di Informatica E Telecomunicazioni V.Le A. Dona, 6-95 125-Catania-Italy, A Pipeline Parallel Architecture For A Fuzzy Inference Processor, 0-7803-5877-5/00/$10.Q0 0 2000 IEEE

[11] Yu Ning Mei* Chen Jingjindepartment Of Electron Engineering, Xi'an University Of Technology, Xi' An 710048,China, ,An Analog Fuzzy Processor Using Neuron-Mos Technology, 0-7803-85u-X/04/$20.00 ©2004 IEEE

[12] Shubhajit Roy, Chowdhury, Hiranmay Saha Jadavpur University. Published By The IEEE Computer Society A High-Performance Fpga-Based Fuzzy Processor Architecture For Medical Diagnosis, 0272-1732/08/$20.00 G 2008 IEEE.

Kamini Sayam, received the B.E . degrees in Electronics Engineering from R.T.M. Nagpur university in 2007 and M.Tech in VLSI from R.T.M.N.U.