Implementation of Complex Fuzzy Logic Modules with VLSI Approach

A.Y.Deshmukh[†], A.B.Bavaskar[††], Dr.P.R.Bajaj[†††] and Dr.A.G.Keskar[††††]

[†]Asst.Professor, ECE Dept, G.H.Raisoni College of Engineering, Nagpur, India
[††]Research Scholar, G.H.Raisoni College of Engineering, Nagpur, India
[†††]Principal, G.H.Raisoni College of Engineering, Nagpur, India
[††††]Professor & Dean R&D,VNIT Nagpur, India

Summary
The hybrid intelligent systems are gaining popularity due to extensive success of these systems in many real world complex problems. In this paper, the implementation of the complex fuzzy logic modules have been undertaken with VLSI design approach. Complex fuzzy logic is a generalization of traditional fuzzy logic based on complex fuzzy sets. The general scheme of Complex Fuzzy Logic Systems(CFLS) is also discussed here. The various stages of computation of fuzzy inferences is discussed alongwith their implementation. The design of various modules of a complex fuzzy processor like Intersection detector module, Execution module, Rule detection module, Defuzzification module have been carried out and the results are evaluated with logic analyser.

Key words:
Complex Fuzzy logic, CFLS, Fuzzy Inference, VLSI

1. Introduction

The framework of fuzzy logic is unique in its ability to represent subjective or linguistic knowledge in terms of a mathematical model. For this reason, fuzzy logic provides a natural method for constructing systems that emulate human decision-making processes. The Fuzzy Inference Rules maps fuzzy linguist inputs to the fuzzy linguist outputs. The inference rules consist of two parts: Aggregation which resolves multiple preconditions of a rule into one degree of truth using fuzzy logic operators. The Composition part assigns an outcome for each fired rule.

Complex fuzzy logic is a postulated logic system that is isomorphic to the complex fuzzy sets. It is a generalization of traditional fuzzy logic, based on complex fuzzy sets. The concept is analogous to the many-valued logics that are isomorphic to type-1 fuzzy sets. A complex fuzzy logic can be defined by particular choices of the conjunction, disjunction and complement operators[1].

Manuscript received September 5, 2008.
Manuscript revised September 20, 2008.

The inference rules are constructed and "fired" like traditional fuzzy logic. The complex fuzzy sets are characterized by complex-valued membership functions which is the novelty. The range of membership functions is extended from the traditional fuzzy range of [0,1] to the unit circle in the complex plane. It provides a method for describing membership in a set in terms of a complex number.

These properties include basic set theoretic operations on complex fuzzy sets - namely complex fuzzy union and intersection, complex fuzzy relations and their composition, and a novel form of set aggregation - vector aggregation. The properties of complex numbers and complex fuzzy sets helps for providing advantages over traditional fuzzy logic.

Complex Fuzzy Sets:

Complex fuzzy sets are a new development in the theory of fuzzy systems. Here, the membership values are complex numbers, drawn from the unit disc of the complex plane. The definition of the complex fuzzy set allows membership in a set to be specified by a complex number, effectively transforming "membership in a set" into a two dimensional concept. Membership of any element, in a complex fuzzy set, is given by a complex-valued grade of membership of the general form

$$r_S(x).\, e^{j\omega_S(X)} (j=\sqrt{-1}, r_S(x) \in [0,1] \qquad (1)$$

which comprises an amplitude term, $r_S(x)$, and a phase term, $^{\omega_S(X)}$ The amplitude term retains the traditional notion of "fuzziness," i.e. the representation of membership in a set as a value in the range [0,1]. The phase term signifies the assertion of complex fuzzy set theory that, at least in some instances, a second dimension of membership is required. Fig 1 shows a typical complex fuzzy set.

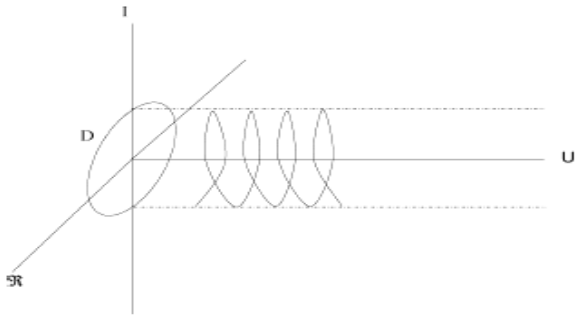

Fig 1 Complex Fuzzy Set

Complex Fuzzy Relations:

The development of complex fuzzy relations and compositions is consequential as it affords the basis for the derivation of complex fuzzy logic. Complex Fuzzy Relations represent both the degree of presence or absence of association, interaction, or interconnectedness, and the phase of association, interaction, or interconnectedness between the elements of two or more crisp sets.

$$R(U,V) = \{((x, y)) \mid (x, y) \in U \times V\} \quad (2)$$

The complex-valued membership function $\mu R(x)=rR(x).ejwn(x)$ is interpreted in the following manner.

$rR(x)$ represents a degree of presence or absence of association, interaction, or interconnectedness between the elements of U and V.
$wR(x)$ represents the phase of association, interaction, or interconnectedness between the elements of U and V.[2-3]

Complex Fuzzy Logic Systems(CFLSs):

Much like the traditional FLS, the CFLS is a nonlinear mapping of an input data vector into an output scalar. A CFLS is characterized by a complex fuzzy rule base containing a set of rules, which are in fact complex fuzzy implications expressed in the form of IF–THEN statements. In essence, a CFLS is a generalization of its traditional counterpart, obtained by replacing the fuzzy sets and fuzzy implications found in a traditional FLS with their complex equivalents. The output of a CFLS is determined in three stages, as illustrated in Fig. 2. The first stage is Fuzzification, used to map crisp inputs into fuzzy input sets. These fuzzy sets may or may not be complex, depending on the application. The second stage, Fuzzy Inference, utilizes a complex fuzzy rule base to map the fuzzy input sets into fuzzy output sets. The complex fuzzy outputs of the separate rules are combined to produce a single complex fuzzy output set.

Defuzzification, as usual, is the final stage of the mapping performed by the CFLS. This stage involves defuzzification of the complex fuzzy output set to produce a crisp output. One possible approach to defuzzification of the complex fuzzy output is to neglect all phase terms and consider only the amplitude term of the output set. Any defuzzification method used in a traditional FLS may be utilized for this purpose. Vector aggregation is in essence a vector sum of its arguments. Therefore, the result of vector aggregation is strongly dependent on the relative phase of its arguments. For example, if all of the arguments are in phase, the amplitude of the sum is maximized. If, however, the arguments are not in phase, the result of the sum may be a grade of membership whose amplitude is smaller than that of its separate arguments. As amplitude is a parameter of consequence in the defuzzification stage, relative phase between inference rules is of great importance. Thus, a form of interaction between rules in a CFLS is defined, requiring that they maintain some sort of phase correlation, or coherence, for a large output grade of membership to be obtained. This interaction between rules is a direct result of the combination of vector aggregation and complex fuzzy implication.

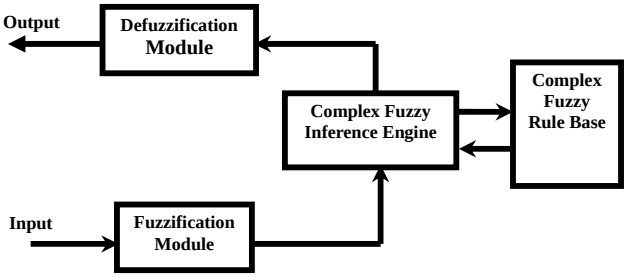

Fig 2 General scheme of a CFLS

2. Methodology

For the design of the fuzzy processor, it is necessary to define the number of input, output, and chained variables. This choice consequently determines the format of the rules, i.e., the maximum number of antecedents and consequents present in a rule. At the same instance, it is necessary to choose the technological process for implementation of the processor. On the basis of this choice one can obtain the maximum clock frequency t_{CK} that is possible to use.

Computation of the fuzzy inference is divided into four pipelined stages:

Stage 1) Acquisition of the inputs and calculation of the vectors which are performed parallel to each other; as the acquisition of each input is independent of the others, several inputs can be acquired at the same time.

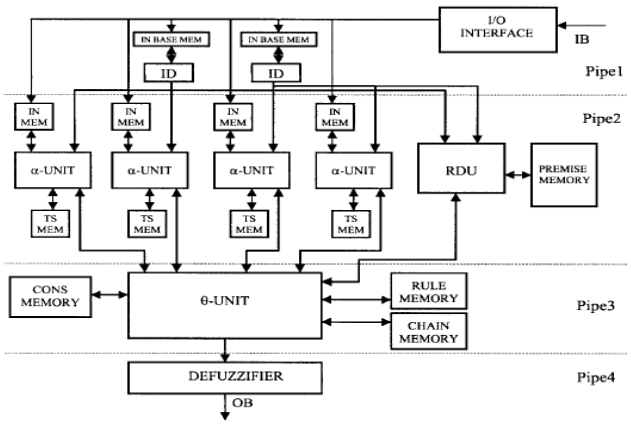

Fig 3 The fuzzy processor

Stage 2) Calculation of the degree of truth of the antecedents and the detection of active rules which are performed in parallel.
Stage 3) Computation of only the active rules with certain degree of parallelism.
Stage 4) Defuzzification of the outputs with parallel calculation of the various outputs.

The hardware architecture of the processor is shown in fig 3. The processor architecture contains the different units like memory unit, intersection detection unit, RDU unit, Execution unit, α unit, θ unit & defuzzifier[4-5].

Each term set memory stores the term sets for a pair of input variables. The number of memories is equal to the number of units. Each input memory stores the fuzzy sets of three input variables. The Cons Memory stores information about the elements in the term sets of the four output variables. Premise memory contains appropriately coded information about the rule premises. Rule memory contains coded information about the antecedents and consequents of the fuzzy rule[6-7].

Intersection Detection Unit:

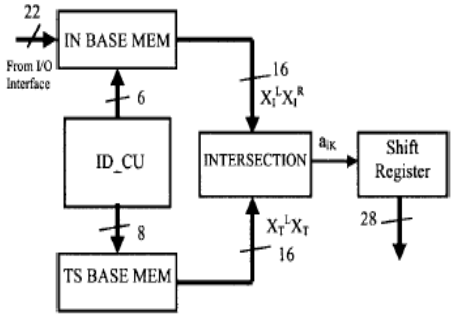

Fig 4 Intersection Detector Bloscк

Fig. 4 shows a block diagram of Intersection Detector (ID) unit. The fig 5 shows the implementation of Intersection Detector unit which detects the vectors associated with the fuzzy inputs[4].

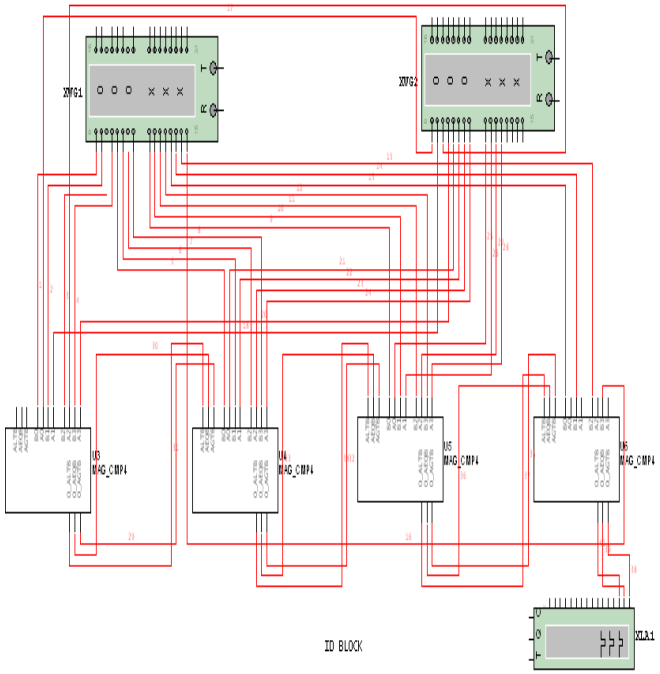

Fig 5 Intersection Detector circuit

It is sufficient to calculate the intersection between the interval made up of the support of the input Xi and that of the fuzzy set F_{ik} of the variable in order to determine the bits $\alpha_i k(K = 1,\ldots n_T)$.

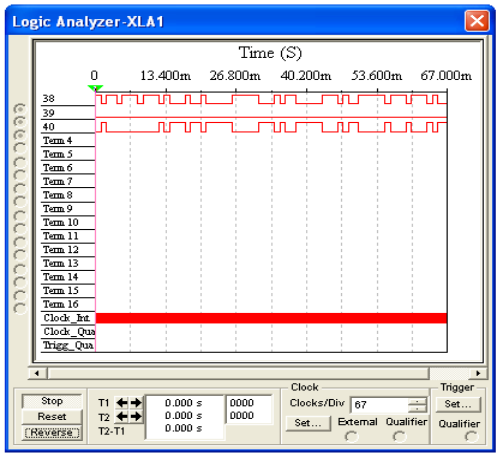

Fig 6 Results of Intersection Detection (ID) unit

Execution Unit:

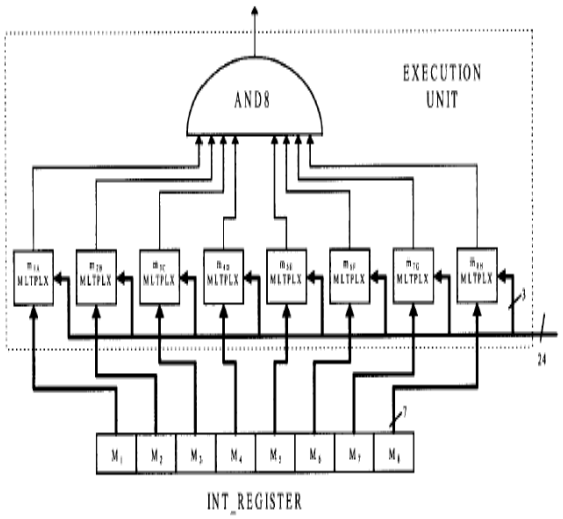

Fig 7 Execution unit block diagram

The fig 7 shows the block diagram of Execution unit and fig 8 shows the implementation of the same. The function of this unit is to generate the active pulse to the rule detection unit. The execution unit is made up of 8 mux with the 8 input AND gate. On the basis of the code the selector produces an output vector of 8 bits, each of which indicates whether the corresponding antecedent has a positive or null degree of activation.

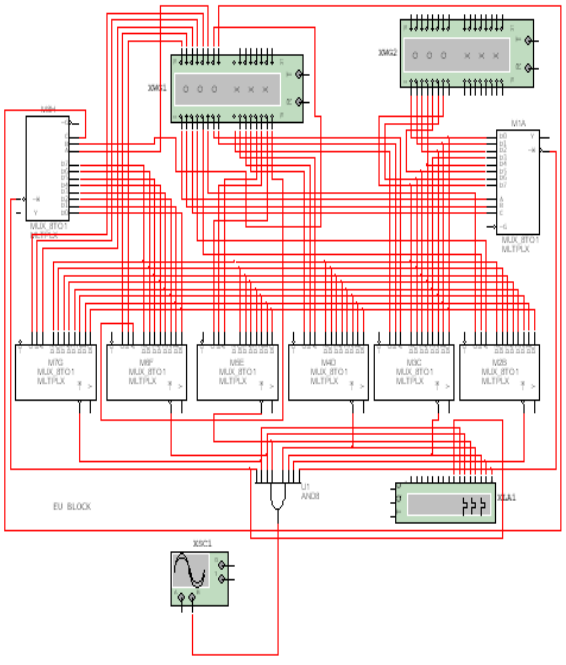

Fig 8 Execution unit block diagram

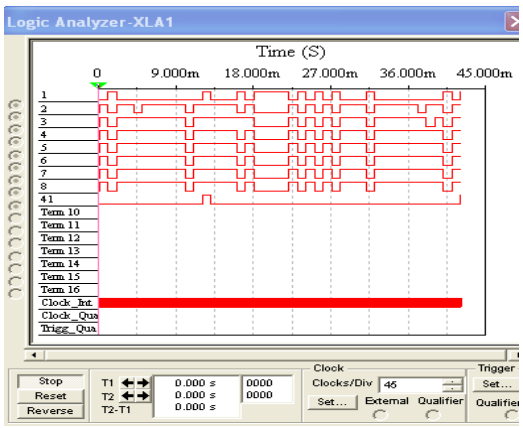

Fig 9 Waveforms of execution unit

Fig 9 shows the waveforms of execution unit with the help of logic analyzer. It shows the generation of active pulse which is required for the Rule Detection Unit.

RDU Block:

The Rule Detection control unit (RD CU) has the task of coordinating the various RDU blocks. On the basis of the information contained in the pointer memory, it feeds the Active Rule Selection block (ARS) with the codes of the premises stored in the premise memory. To increase the speed of the RDU the activities of the ARS block are pipelined with the search in the pointer memory for the next group of rules to be processed[4].

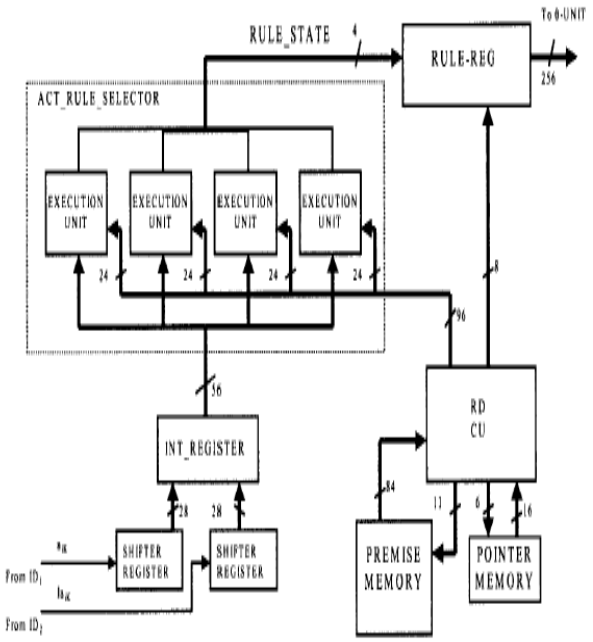

Fig 10 RDU unit Block diagram

This unit detects all the active rules to be processed by the θunit. Fig. 10 & 11 shows the block diagram & the implementation of the Rule Detection Unit(RDU). The pointer memory stores a table, the number of entries in which is equal to the number of groups stored in the premise memory. The kth entry contains the address of the first rule in the group and the number of rules it contains.

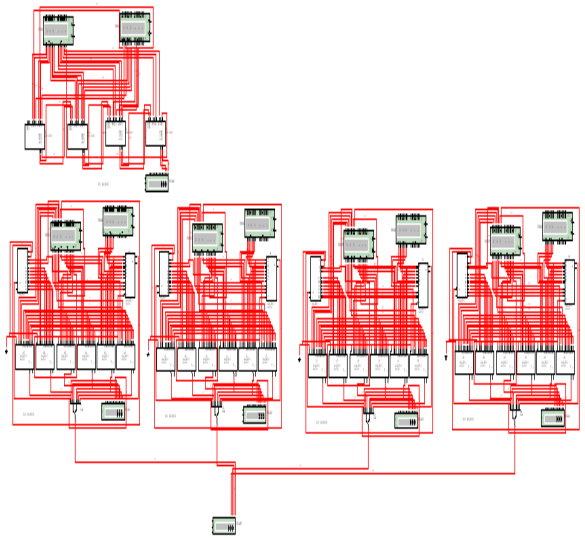

Fig 11 Implementation of RDU unit circuit

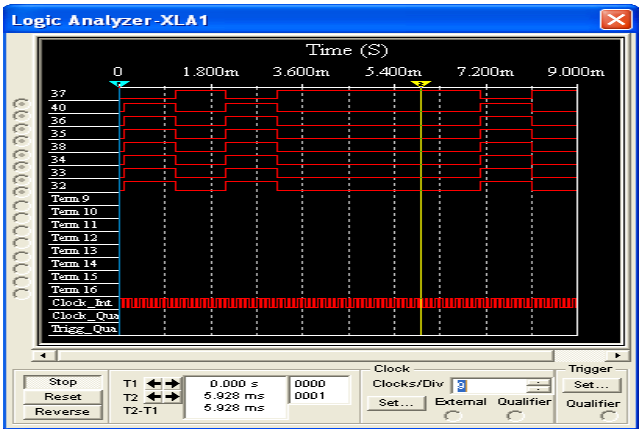

Fig 12 Output of Rule Detection Unit with Logic Analyser XLA1

The fig 12,13,14 shows the outputs of Rule Detection unit. The output of execution unit and intersection detector is applied to the rule detection unit.

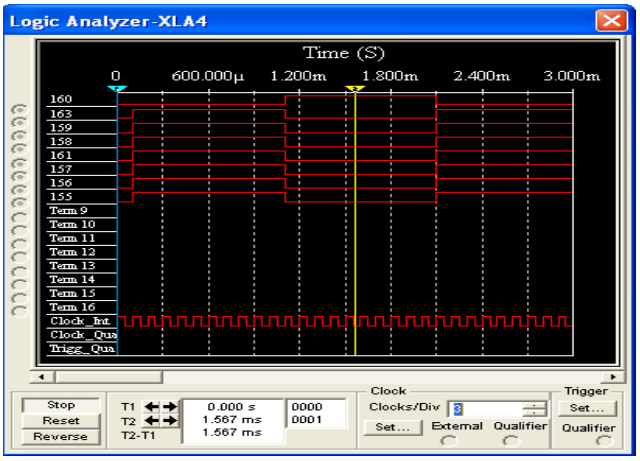

Fig 13 Output of Rule Detection Unit with Logic Analyser XLA4

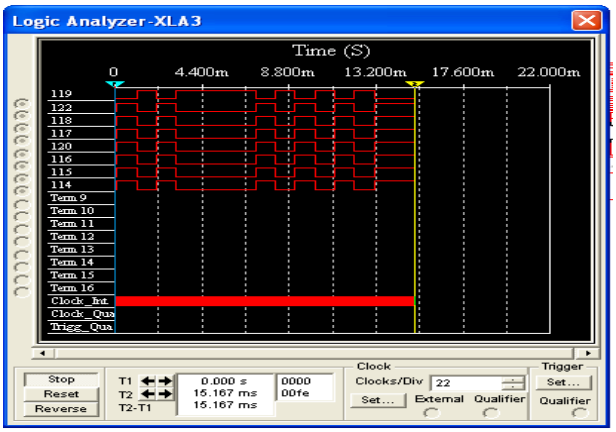

Fig 14 Output of Rule Detection Unit with Logic Analyer XLA3

Defuzzifier Unit:

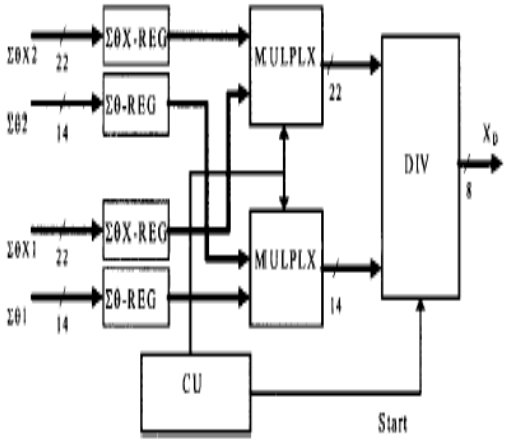

Fig 15 Defuzzifier unit Block diagram

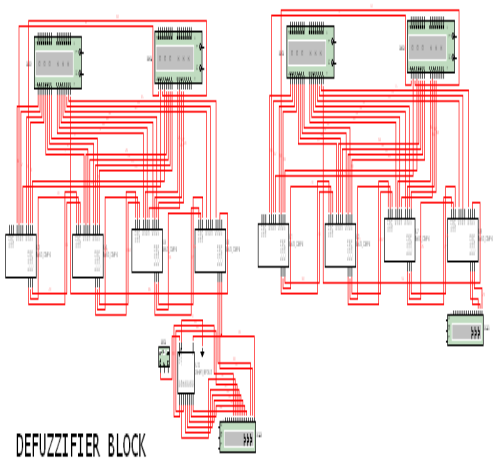

Fig 16 Defuzzifier unit circuit

The fig 15 & 16 shows the block & circuit diagram of defuzzifier unit. A single divider is sufficient to compute the defuzzified value. As soon as the sums are stored in the registers, the divider starts to operate. Its input is the outputs of the two multiplexers. The beginning of the division is signaled by a start signal, which is activated by the CU of the θ-unit as soon as the input data is ready[4].

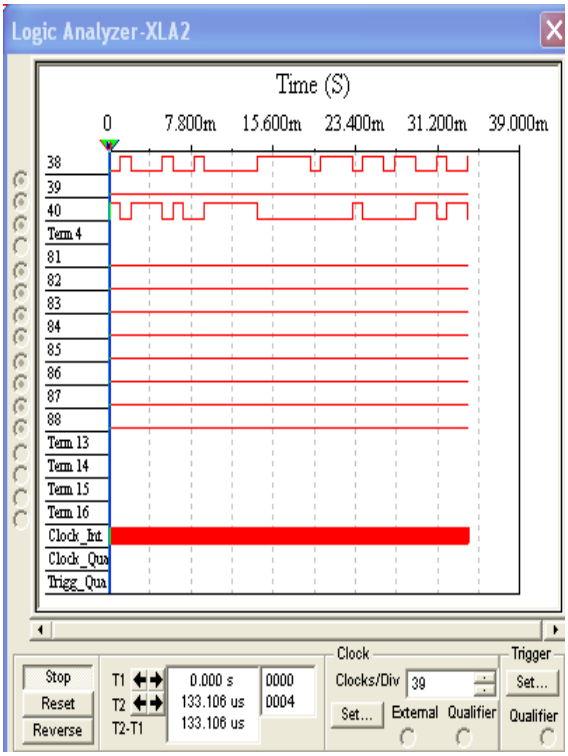

Fig 17 Output of Defuzzifier circuit

Fig 17 shows the output of Defuzzifier circuit which is obtained by using the logic analyzer. The output of the defuzzifier is the divider output.

3. Conclusion

The various modules related to complex fuzzy logic have been designed. Using this, it is possible to realize a complex fuzzy system with the VLSI approach. With the prior mention of number of input, output, and chained variables, the design of fuzzy processor can be carried out. With the help of acquisition of the inputs and calculation of the vectors, calculation of the degree of truth of the antecedents and the detection of active rules & defuzzification of the outputs, the fuzzy inferences can be computed. Significant reduction in number of rules can be achieved by the detection process.

References

[1] D. Ramot, M. Friedm D. Ramot, M. Friedman, G. Langholz, and A. Kandel, "Complex fuzzy logic," *IEEE Trans. Fuzzy Syst.*, vol.11, no. 4, pp. 450–461, Aug. 2003.

[2] D. Ramot, R. Milo, M. Friedman, and A. Kandel, "Complex fuzzy sets," *IEEE Trans. Fuzzy Syst.*, vol. 10, pp. 171–186, Apr. 2002.

[3] Scott Dick, "Toward Complex fuzzy logic," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 3, pp. 405–414, June. 2005.

[4] Giuseppe Ascia, Vincenzo Catania, and Marco Russo, "VLSI Hardware Architecture for Complex Fuzzy Systems", *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 5, October 1999.

[5] G. Ascia, V. Catania, D. Panno, G. Ficili, and S. Palazzo, "A VLSI fuzzy expert system for real-time traffic control in ATM networks," *IEEE Trans. Fuzzy Syst.*, vol. 5, pp. 20–31, Feb. 1997.

[6] G. Ascia, V. Catania, M. Russo, and L. Vita, "Rule driven VLSI fuzzy processor," *IEEE Micro*, vol. 16, pp. 62–74, June 1996.

[7] G. Ascia, V. Catania, "An efficient hardware architecture to support complex fuzzy reasoning," *ictai*, p. 250, Seventh International Conference on Tools with Artificial Intelligence, 1995

A.Y.Deshmukh received the B.E. degree from YCCE Nagpur and M.E. degree from SGGS Nanded, India in 1996 and 2001, respectively. He is pursuing Ph.D. from VNIT Nagpur. He has published around 20 research papers and attended International conferences at Australia & USA. He worked as Program Co-Chair & Publication Chair for 1st International Conference on Emerging Trends in Engg & Technology(ICETET-08). He has worked as reviewer for many International & National conferences. He is a member of IEEE, CSI, ISTE. His research areas include VLSI, Hybrid Intelligent Systems, Soft Computing etc. He is also receipient of Best Teacher Award for 2004-05.

A.B.Bavaskar is a research scholar at GHRCE, Nagpur, India. He is also Senior R & D Engineer at ARC-TEC SYSTEM LIMITED, Nagpur. His research areas include VLSI, Complex Fuzzy Systems.

Dr.Preeti R.Bajaj holds a PhD degree in Electronics Engineering. Currently she is working as Principal, GHRCE, Nagpur, India. Her research interests include Soft Computing and optimizations, Hybrid Systems & Applications of Fuzzy logic in areas of Intelligent Transportation Systems. Her professional societies affiliation include Member- Institute of Engineers, Senior Member- IEEE, LM-ISTE, ILM-UK, LM-CSI. She has chaired many technical sessions at International Conferences in India and abroad. She is also a recipient of Best Teacher Award in the year 2001-02. She worked as reviewer for many International Conferences and Journals in India and abroad. She also worked as General Chair for First International Conference on Emerging Trends in Engineering & Technology (ICETET-08) at GHRCE Nagpur, India. She has over 30 publications in refereed International Conferences and Journals.

Dr.A.G.Keskar is working as Professor & Dean R&D at VNIT, Nagpur, India. He is recipient of Gold Medal at BE & ME. He has 20 years of Teaching & 7 years Industry experience. He has published many research papers and attended International conferences at abroad. He has completed an Industrial Project at Rotterdam, UK. He has undertaken many consultancies. He is Senior Member IEEE, FIETE, LMISTE, FIE. He has guided many Ph.D. candidates. His research areas include Embedded Systems, Fuzzy logic, Soft computing etc.