Designing a Smart XOR Gate Depending Kenevan Truth Interval in Fuzzy Logic

Asst. Prof. Dr. Muna Hadi Saleh
Baghdad University/College of Education for Women
Computer Science Department

Abstract:

The new Kenevan truth interval fuzzy logic, in which truth values of propositions are represented as subintervals of the real unit interval that contain the single truth value rather than the truth value itself, is described here. The classical propositional logic is shown to be a special case of the truth interval fuzzy logic in which the truth intervals are restricted to [0.0, 1.0]. This paper has presented both a technique and development environment for designing and the production of hardware suitable fuzzy logic for some specific applications. This technique a makes set of a fuzzy data, ensuring that a high degree of both flexibility and programmability is obtained with minimal hardware complexity. Here, some new important properties of Kenevan fuzzy XOR operation were introduced, that made it possible to represent the logic formula in (KFXOR) form. Finally this circuit is designed and implemented by computer. Simulation and experimental results of (KFXOR) are presented.

***Keywords:** Fuzzy logic, Kenevan truth interval fuzzy logic, Inference Rule Union, Intersection, KFOR gate, KFAND gate, FXOR gate.*

تصميم بوابة (أو إلاستثنائية) الذكية بالاعتماد على فترة كينيڤان الحقيقية في المنطق المضبب

أ.م. د. منى هادي صالح
جامعة بغداد/ كلية التربية للبنات / قسم علوم الحاسبات

المستخلص:

القيم الحقيقية لفرضيات كينيڤان لفترة المنطق المضبب الحقيقية الجديدة تتمثل لفترات جزئية من وحدة فترة الحقيقية، والتي تتضمن قيمة حقيقة مفردة عوضاً عن القيمة الحقيقية ذاتها. حيث تعد الفرضية القديمة حالة خاصة من فترة المنطق المضبب الحقيقية. قدم في هذا البحث، التقنية والتطور اللازمين لتصميم الانتاج المادي للمنطق المضبب لبعض التطبيقات

المتخصصة. فهذه التقنية تؤكد الحصول على أعلى درجة المرونة البرمجية لمجموعة البيانات المضببة، مع أقل تعقيد مادي، بعض الخصائص الجديدة لعمل بوابة "أو" الإستثنائية، التي تمكن من وضع صياغة هيكلة خاصة ببوابة "أو" الأستثنائية تم عرضُعها وأخيراً، تم تصميم بوا بة وتنفيذها "أو" الإستثنائية بوساطة الحاسوب من خلال إستعمال حقيبة برمجية خاصة لتصميم الدوائر الالكترونية وتنفيذها كما تم تشبيها وعرض النتائج التجريبية لها.

1- Introduction

Fuzzy Logic, introduced by Zadeh, has proven to be a powerful means of problem solving. It has gained success in various disciplines, among others, control, decision theory, operations research, and management science. According to Zimmermann the characteristics of reality establish the aptitude and the success of fuzzy logic as a means of describing and modeling reality [1]. Real situations are often not crisp and cannot be described precisely. Moreover, the description of a real system often requires far more data than can be processed. Since fuzzy logic reflects this natural fuzziness well it has not only become an established part of the mentioned disciplines but has also attained considerable economical relevance. For the implementation of fuzzy logic two options are available: the software and the hardware approach. The software approach simply consists of programming the demanded functions by use of any programming language [2].

The main drawback is the lack of programming facilities, a property that has lead to the pre-eminence that the digital realizations have today. Accordingly, analog implementations are generally used as tailored solutions for applications with highest speed requirements or, exploiting the comparatively low circuit complexity, as inexpensive solutions for high-volume products [3][4]. The mentioned drawback is overcame by the promising new approach of fuzzy logic circuits, which is supplemented by digital circuitry for programmability.

A special software tool is usually interfaced to support problem formulation in terms of fuzzy logic. The software approach is distinguished by low cost and high flexibility, permitting quick and easy alterations. It is well suited for prototyping and a large number of applications. However, the serious disadvantage is the low

execution speed of the fuzzy logical connectives, which has proven to be far too low for a lot of important applications such as embedded control systems. The demand for high-speed execution of the fuzzy inference has made the hardware approach attractive [3][5].

It has been pursued from the beginning of the fuzzy logic boom in the mid of the eighties. The characteristic of fuzzy logic hardware is that basic fuzzy logic connectives are implemented with either digital or analog dedicated circuitry. The architecture of such circuits is tailored to the fuzzy inference, which is the essential operation to be executed, due to the knowledge-based nature of the fuzzy logic approach of problem solving [6]. Fuzzy logic hardware can be subdivided into fuzzy coprocessors and general fuzzy processors, which can operate stand-alone. Fuzzy logic circuits are not only a subject of intensive research but have also found access to the market. Several fuzzy processors are available today and fuzzy logic circuits find an application in industrial processes as well as in consumer products. Due to increasing complexity and speed requirements of the applications, the hardware share among the fuzzy logic applications is predicted to rise rise up to 30% in 1997 [7].

Fuzzy logic's basics of being subjective make it applicable to many fields. Implementing it in systems is fairly simple with flexible software solutions and fast fuzzy processors. It has the potential to provide many businesses with industrial applications as well as commercial products. It is not hard to see the advantages to using fuzzy logic solutions, and it is quite evident how important it will become. In the future, it will probably be very difficult to find a system sans fuzzy logic in some form or fashion. The ordinary fuzzy hardware system, especially using a digital technique, is constructed by the fuzzy logic gates circuits [8]. The paper is organized as follows: the principle of Kenevan truth interval fuzzy logic and suitable propositions has been given.

The special characteristics of KFXOR and of the fuzzy logic gates are reported. Finally, the logic circuits which are composed of simple electronic tool have been designed.

2- Related Works

The fuzzy logic circuit designing strategy started nearly a decade back. In 1995(Pérez and Bañuelos) present electronic models

of fuzzy gates that satisfy the fundamental principles of fuzzy logic. (Stefan [9]) discusses fuzzy logic, including how and why it is applied in various situations, and detailing how it is applied through software and hardware means. Finally in 2009 (Andrew) discusses fuzzy logic, including how and why it is applied in various situations, and detailing how it is applied through software and hardware means. The idea of this work is to exploit this relationship for testing. For fault modeling the behavior of faulty fuzzy gates is investigated through simulations of transistor-level failures such as shorts and opens. A slightly modified five-valued logic which allows for the analog test responses makes it possible to use of the test methods for digital circuits [10].

3- Fuzzy Truth Interval Logic

In his 1973 paper, Lotfi Zadeh introduced what is now called fuzzy logic on fuzzy control systems and proceeded to refine its definition in subsequent papers. Zadeh presented the truth values for propositions of his fuzzy logic as any real number in the inclusive unit interval, or [0.0, 1.0] [11]. A proposition "A" with truth value "a" is written as "A:a" given this convention, the fuzzy logic compositional rules formulated by Zadeh are presented below as Axioms (1, 2, and 3):

Axiom 1 $A: a \wedge B: b := (A \wedge B): \min(a, b)$
Axiom 2 $A : a \vee B : b := (A \vee B) : \max(a, b)$
Axiom 3 $\neg(A, a) : A : (1 - a)$

These compositional rules clearly include those of the classical two-valued logic as a special case. Axioms (1-3) give an algorithm for combining truth values of propositions to get the truth value of a composite proposition. It should be using the max and min operations extensively and, it proposes and uses the following notation for these operators [12]:

abc … z = max (a, b, c, …, z),
abc … z = min (a, b, c, … , z).

The single truth value designated in the above axioms is assumed to exist, will be represented as a sub-interval of the closed unit interval which includes the postulated single truth value. "A:a" will, therefore, be represented as A:[a_0, a_1] where the interval [a_0, a_1] is assumed to contain "a". One reason for representing truth values in this manner is that, in practice, the actual truth value is seldom known exactly. It is simply more convenient to represent the truth value as an interval than to guess at an exact value for the purposes of computation [13]. A second, more compelling, reason for the expression of the truth values as subintervals of the unit truth intervals in the increased power of exposition is attendant upon by this choice. Using a single valued notation an unknown truth value can only be represented as an unknown, i.e., "A:x".

Using a fuzzy interval logic, an unknown truth value is simply represented as the unit interval itself, i.e., A:[0.0,1.0]. A proposition "A" in Zadeh's dispositional logic (a disposition is a proposition with truth value of probably true) can be represented in the Kenevan truth interval fuzzy logic as A: [x_0, x_1], $0.5 < x_0 \leq x_1 \leq 1.0$ [14].

4- Kenevan Truth Interval in Fuzzy Logic

The definition of the Kenevan truth interval fuzzy logic, originated and described by James Kenevan in [11], assumes that every proposition has a single truth value contained in the closed unit interval of real numbers [0.0, 1.0]. The value "0.0" is considered to be the classical truth value of false, while "1.0" is the classical truth value true. If the single truth value for the fuzzy propositions "A" and "B" is known to be "a" and "b", respectively, then the model, or function "µ" that maps propositions onto the set of truth values [0.0, 1.0], is defined as follows [11][13][14]:

Definition 1 $\mu(A) \rightarrow a, \quad a \in [0.0, 1.0].$

Definition 2 $\mu(B) \rightarrow b, \quad b \in [0.0,1.0].$

Definition 3 $\mu(A \wedge B) \rightarrow \min(a, b) = \underline{ab}.$

Definition 4 $\mu(A \vee B) \rightarrow \max(a, b) = \overline{ab}.$

Definition 5 $\mu(\neg A) \rightarrow (1 - a).$

Note that if "a" and "b" are limited to the classical truth value analogs of true and false, "1.0" and "0.0" respectively. The design generates the same truth values for the connective ($\wedge$, $\vee$, and $\neg$) as does the classical definition. The definition of (µ) given above for the Kenevan Logic, then, includes the classical definition as a special case. Even though it assume, that a single truth value for each proposition exists, designate it, not by a single element of [0.0, 1.0], but by the subinterval of [0.0, 1.0] that contains it. If the truth value is unknown, it is then represented as [0.0, 1.0]. If the precise truth value of a proposition "A" is "a", but "a" is not known exactly, the proposition "A" is then written as A:[a_0, a_1] where $0.0 \leq a_0 \leq a \leq a_1 \leq 1.0$. If the truth value of "A", "a", is known precisely this fact can be expressed by writing A:[a, a].

The Kenevan truth interval fuzzy propositional logic suffers from the same limitation as the classical propositional logic, i.e., requiring separate sentences to express the truth values of the same property applied to different objects. This limitation can be removed by representing objects as variables and writing A(x) to mean that the object "x" has the property "A". The single truth value of this statement, which clearly depends on the identification of the object "x", would then be designated by A:a(x), or, using a truth interval, as A:[$a_0(x)$, $a_1(x)$].

As a general schema for a predicate in the Kenevan truth interval fuzzy predicate logic it has $A(x_1, x_2, x_3, \ldots, x_n)$:[$a_0 (x_1, x_2, \ldots, x_n)$, $a_1(x_0, x_2, \ldots, x_n)$], where the "$a_i$" are n-array functions a:D_n[0.0, 1.0] which associate each predicate with a sub-interval in the closed unit interval and are the fuzzy equivalent of the mappings that, as part of a classical interpretation, map predicates onto the two element set true and false. The basic inference rules for the propositional version of the Kenevan truth interval fuzzy logic, taken from [12], are given below. The first inference rule is that for disjunction, which, like its classical counterpart, involves two possible alternatives.

4.1 Inference Rule for OR ($\vee$):

To emphasize the fact that each leg of the table below represents a different alternative, we may refer to each leg as describing a different world, as is customary in modal logic. The inference rule for conjunction, which, unlike its classical equivalent, generates alternative worlds, is given below:

$$A: [a_0, a_1]$$
$$B: [b_0, b_1]$$
$$A \vee B: [p_0, p_1]$$

$A : [a_0, p_1a_1]$	$A : [p_0a_0, p_1a_1]$
$B : [p_0b_0, p_1b_1]$	$B : [b_0, p_1b_1]$
$A \vee B : [p_0a_0b_0, p_1a_1b_1]$	$A \vee B: [p_0a_0b_0, p_1a_1b_1]$

4.2 Inference Rule for AND ($\wedge$):

The inference rules for intersection follow:

$$A: [a_0, a_1]$$
$$B: [b_0, b_1]$$
$$A \wedge B: [q_0, q_1]$$

$A : [q_0a_0, a_1]$	$A : [q_0a_0, q_1a_1]$
$B : [q_0b_0, q_1b_1]$	$B : [q_0 b_0, b_1]$
$A \wedge B : [q_0a_0b_0, q_1a_1b_1]$	$A \wedge B: [q_0a_0b_0, q_1a_1b_1]$

4.3 Inference Rule for NOT ($\neg$):

The inference rules for negation follow:

$$A: [a_0, a_1]$$
$$A: [1 - a_1, 1 - a_0] \neg$$

5- Design Analog Fuzzy Logic Circuits

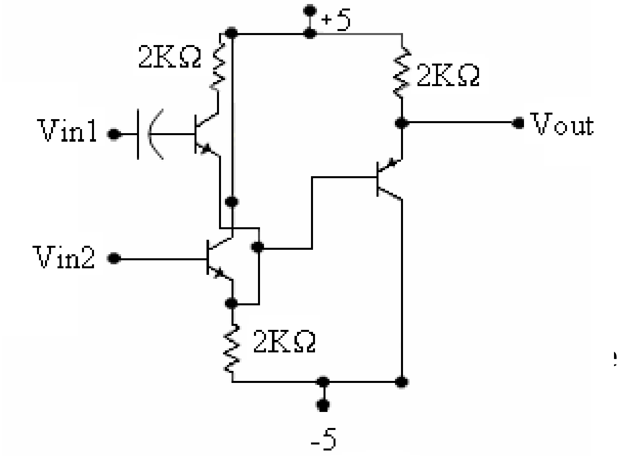
Analog fuzzy logic circuits offer highest speed without consuming much chip area. This is due to three factors: First, no time-consuming Analog-to-Digital (A/D) and Digital-to-Analog (D/A) conversion of the processed physical quantities are necessary. Second, a fuzzy logical value can be represented by the voltage or current of one signal instead of a number of bit lines. Third, simple analog

circuits exist for the implementation of the basic fuzzy connectives, making fully parallel architectures and data processing possible. In this work, the class of analog fuzzy logic circuits, which can be regarded as generalization of digital logic circuits to a continuous range of logical values is dealt with. Since these logical values are represented by the voltage or current level of one signal, the circuits are analog [4][8]. Hence it becomes clear that, on the one hand, the concepts of digital testing cannot be applied directly. The functional concepts for testing analog circuits, on the other hand, appear not suitable because they do not allow the special structure of analog fuzzy logic circuits, which is the same as that of digital circuits. An embodiment of the present work provides a logical fuzzy union and intersection operation circuit which is simple and uses a traditional architecture generally dedicated to the arithmetic/logical operations.

Designing these circuits includes simple components (transistor, OP-AMP "Operational Amplifier", and others), the specifications of these components have been taken from [15]. There are two levels, the first level for the selection of the logical fuzzy union operation and the second level for the selection of the logical fuzzy intersection operation. A major issue here is how to generalize each stage of the process to make the chip flexible enough to handle many different systems. This is necessary since reprogramming will be possible, and designing a chip for every different system is flexible. In order to develop this logic electronically constructs the basic gates of the logic: "KFAND", "KFOR", and "KFXOR".

5.1 Kenevan Fuzzy Union Circuit (KFOR-Gate):

The Kenevan fuzzy union operation can be represented by using definition 4 in section 3, [Y= $\mu(A \vee B) \rightarrow \max(a,\ b) = \overline{ab}$ **], where "Y" is the output and inputs (x_1,x_2) both are in [0.0V, 5.0V] voltage level. The inputs and outputs values are in volts corresponding to the interval [$0.0 \le a_0 \le a \le a_1 \le 1.0$]. The electronic design of the KFOR gate is illustrated in figure (1) below: -**



5.2 Kenevan Fuzzy Intersection Circuit (KFAND-Gate):

The Kenevan fuzzy intersection operation can be represented by using definition 3 in section 3, [Y= $\mu(A \wedge B) \rightarrow \min(a,\ b) = \underline{ab}$], where "Y" is the output and "x_1" is a first input and "x_2" represent the second input, both are in [0.0V, 5.0V] voltage level. Remember that the gate "KFAND" always takes the minimal value of the two inputs in $\min(x_1, x_2)$. The input and output values in volts corresponding to the interval [$0.0 \le a_0 \le a \le a_1 \le 1.0$]. The electronic design of the KFAND circuit has been shown in figure (2) below: -

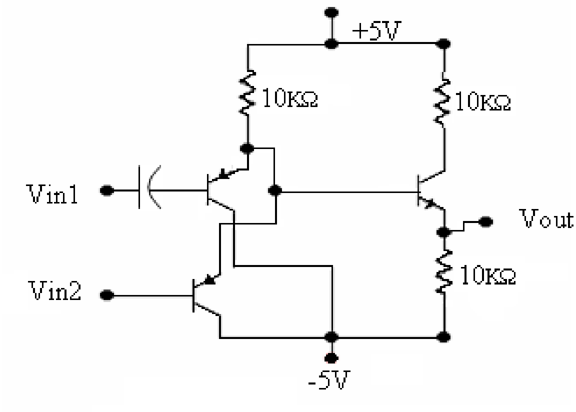


Figure (2): Kenevan Fuzzy and (KFAND) Gate

5.3 Kenevan Fuzzy Exclusive-OR Logic (KFXOR-Gate):

The common fuzzy logic approach of problem solving is a knowledge-based one. This is in particular true for fuzzy logic hardware, which is essentially the realization of an expert system. The crucial point is the minimization of the execution time for the complicated operation of inference making by suitable smart circuit architecture. The main advantages of using KFXOR logic gates are:-

1- Many are useful for KFXOR structure such as arithmetic, parity check and circuits.
2-KFXOR based circuits require smaller layout areas for their realization on very large scale integration (VLSI) chips.
3-KFXOR circuits are less expansive; it improves testability and reduces the total design.

A better understanding of the KFXOR will now be described, which contains a circuit for performing logical fuzzy union and intersection operations according to the KFXOR. Circuit simulation based on fuzzy logic and Kenevan truth interval is an alternative to the more usual Boolean-logic analysis. The operation of "KFXOR" gate governed by the relation; $Y= \max [\min(x_1, (1-x_2)), \min((1-x_1), x_2)]$, Where (Y) is the output and "x_1, x_2" are two inputs. The architecture dedicated to the arithmetic/logical operations and so it may be employed in simple manner in any digital device. In particular, the present smart circuit has an extremely simple architecture formed by conventional components as shown in figure (3). It's possible to identify a whole mapping in the interval $[0.0 \le a_0 \le a \le a_1 \le 1.0]$ with a KFXOR logic formula.

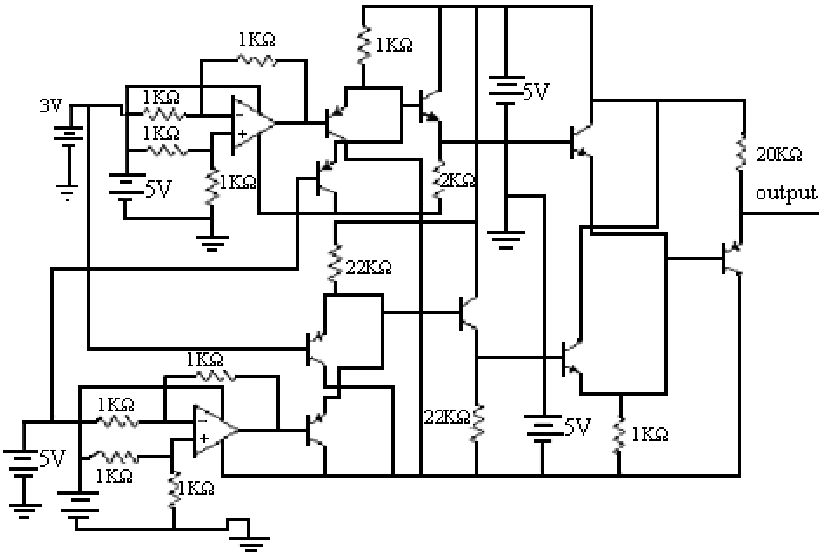


Figure (3): Kenevan Fuzzy XOR (KFXOR) Gate

6- Conclusion and Simulation Results:

Real implementation of the complementation, min and, max circuits have been proposed. The proposed design are very easy to understand and implemented, which are direct and fast techniques for computing the grade membership of KFXOR, when compared with the traditional design. This smart circuit is likely to become the future hardware solution for high-performance fuzzy logic applications. In these circuits, fuzzy values [0.0, 1.0] are indicated in voltage level [0.0V, 5.0V], and made by using a transistor-level. Then, investigated and simulated these fuzzy circuits through the electronics software as illustrated in figure (4). It can summarize the results of KFXOR gate in table (1), which will show the input and output values in volts corresponding to the interval [0.0, 1.0]. Although, the results are considered good, but, there is some a disadvantage which must be mentioned; that the design becomes complex when one increases the number of inputs.

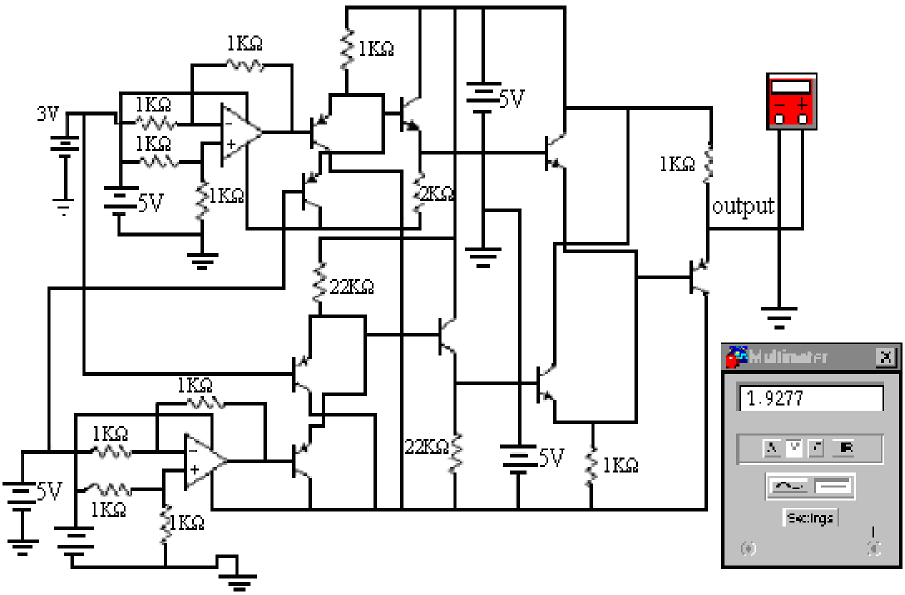


Figure (4): Investigated KFXOR Gate

Table (1): Results of KFXOR Gate

I/P value (Kenevan truth interval Fuzzy)				corresponding I/p value (volts)				Kenevan Fuzzy O/P value	corresponding O/P value in (volts)
x_1	x_2	$\bar{x}_1$	$\bar{x}_2$	x_1	x_2	$\bar{x}_1$	$\bar{x}_2$	Y	Vout
0	0.9	1	0.1	0	4.5	5	0.5	0.9	4.5
0.3	0.2	0.7	0.8	1.5	1	3.5	4	0.3	1.5
0.7	0.5	0.3	0.5	3.5	2.5	1.5	2.5	0.5	2.5
0.9	0.1	0.1	0.9	4.5	0.5	0.5	4.5	0.9	4.5
1	0.3	0	0.7	5	1.5	0	3.5	0.7	3.5
0.2	0.4	0.8	0.6	1	2	4	3	0.4	2

References

[1]Chellas B., "Modal Logic an Introduction," Cambridge University Press, Cambridge, 1980.

[2]X. Zahiwei, "Multivalued Logic and Fuzzy Logic their Relationship, Minimization, and Application to Fault Diagnosis," IEEE Transaction Computer, Vol. 33, July, 1984.

[3] Hung et al, "Implementing a Fuzzy Inference Engine Using FPGA," Sixth Annual IEEE International ASIC Conference and Exhibit Proceedings; Sep. 27-Oct. 1, 1993.

[4] Pérez J. L., M. A. Bañuelos, "Electronic Models of Fuzzy Gates," Instrumentation and Development, Vol. 3, No. 5, 1995.

[5] Watanabe et al., "Evaluation of Min/Max Instructions for Fuzzy Information Processing," IEEE Transactions on Fuzzy Systems Vol. 4, No. 3, 1996.

[6] Robert B., "Data-Driven Fuzzy Modeling: Transparency and Complexity," Delft University of Technology, 2004.

[7] Giacalone et al; "Hardware implementation Versus Software Emulation of Fuzzy Algorithms in real applications," the IEEE International Conference on Fuzzy Systems Proceedings, Vol. 1, May, 1998.

[8] Andrew Z. McCord, "Fuzzy Logic and Its Applications in Hardware," March 4, 2009.

[9] Stefan Weiner, "A Fault Model and a Test Method for Analog Fuzzy Logic Circuits," International test conference, 1995, Paper 9.4, pp. 282-291

[10] Andrew Z. McCord, "Fuzzy Logic and Its Applications in Hardware," for senior seminar, CPSC 491-01, March 4, 2009.

[11] Kenevan J.R., R.E. Neapolitan, "A Model Theoretic Approach to Propositional Fuzzy Logic using Beth Tableau, in: L, Zadeh, J. Kacprzyk (Eds.)," Fuzzy Logic for the Management of Uncertainty, Wiley, New York, NY, 1992.

[12] M. Mukaidono, "Kleene Algebra's in Fuzzy Truth Values," The Fourth Inter. Workshop on Rough Sets, Fuzzy Sets and Machine Discovery, University of Tokyo, Nov., 2005.

[13] M. Mukaidono, "Algebraic Structures of Interval Truth in Fuzzy Logic," FUZZ-IEEE Proc.,Vol II, 2007.

[14]Carl W. Entemann, "A Fuzzy Logic with Interval Truth Values," 430 South Michigan Avenue, Chicago, USA, March, 2000.

[15] "RS Components Data Sheet," Texas Instruments Company, 1998.