# Generators of Fuzzy Operations for Hardware Implementation of Fuzzy Systems

Imre J. Rudas[1], Ildar Z. Batyrshin[2], Antonio Hernández Zavala[3], Oscar Camacho Nieto[3], László Horváth[1], and Luis Villa Vargas[3]

[1] Budapest Tech
rudas@bmf.hu, horvath.laszlo@nik.bmf.hu
[2] Mexican Petroleum Institute
batyr1@gmail.com
[3] Mexican Polytechnic Institute – Computer Research Centre
antonioh@hotmail.com, {oscarc,lvilla}@cic.ipn.mx

**Abstract.** The problem of effective hardware implementation of parametric operations of fuzzy systems is studied in this paper. The methods of generation of parametric classes of fuzzy conjunctions and disjunctions by means of introduced generators and basic operations are considered. Several types of generators of parametric fuzzy operations simple for hardware implementation are proposed. Examples of hardware implementation of proposed parametric operations are presented.

**Keywords:** Fuzzy logic, fuzzy processor, fuzzy system, conjunction, generator.

## 1 Introduction

Hardware implementation of fuzzy systems plays an important role in many industrial applications of fuzzy logic based intelligent systems in control systems, signal processing, pattern recognition, expert systems, decision making etc [1-3]. To provide a flexibility of fuzzy systems in modeling real processes it is important to implement families of parametric fuzzy logic operations that can be tuned to obtain better simulation results of fuzzy system [4-6]. On-board and real-time applications of fuzzy systems require faster processing speed of fuzzy hardware. To achieve these goals it is desirable to propose fuzzy parametric operations that can be effectively implemented in hardware. In [7], a new method of generation of parametric family of fuzzy conjunctions that have simple hardware implementation was proposed and FPGA digital implementation of new parametric conjunctions using Xilinx tools was considered.

In this paper we propose a new approach to generation of parametric families of fuzzy conjunctions simple for hardware implementation. This approach is based on the use of simple parametric generators together with *min-max*, Lukasiewicz and drastic *t*-norms and *t*-conorms [8-9] that have effective digital implementation. Several simple parametric classes of generators are proposed and examples of parametric conjunctions generated by means of these generators and basic *t*-norms and *t*-norms are considered. The proposed approach is based on the results of papers [4-6] where

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 710–719, 2008.

the methods of generation of parametric classes of fuzzy operations simple for tuning in fuzzy systems were introduced. But most of parametric operations proposed in [4-6] use product operations or exponential functions that do not admit efficient hardware implementation. In this paper we introduce new parametric families of fuzzy operations that have efficient digital hardware implementation.

The paper has the following structure. Section 2 gives the basic definitions of fuzzy conjunction and disjunction operations and considers the methods of generation of these operations. In Section 3 a digital representation of fuzzy operations is considered and several simple parametric families of generators of fuzzy conjunctions are proposed. The methods of a hardware implementation of proposed parametric operations are considered in Section 4. In Conclusions we discuss obtained results and future directions of research.

## 2 Basic Definitions

In binary logic, the set of true values contains only two elements $L = \{0,1\}$, the negation $\neg$, conjunction $\wedge$ and disjunction $\vee$ operations are defined as follows:

$$\neg 0 = 1, \qquad \neg 1 = 0, \tag{1}$$

$$0\wedge 0 = 0,\ 1\wedge 0 = 0,\ 0\wedge 1 = 0, \quad 1\wedge 1 = 1, \tag{2}$$

$$0\vee 0 = 0, \quad 1\vee 0 = 1,\ 0\vee 1 = 1,\ 1\vee 1 = 1. \tag{3}$$

In fuzzy logic, the set of true values (called membership values) usually contains continuum number of elements $L= [0,1]$ and negation, conjunction and disjunction operations are defined as functions $N{:}L \to L$, $T{:}L\times L \to L$ and $S{:}L\times L \to L$. Generalization of properties of this operations (1)-(3) on the set $L= [0,1]$ can be done by different ways [10-12, 6, 5, 8,13].

Fuzzy involutive negation is defined by the following axioms:

$$N(0)= 1, \qquad N(1) = 0, \tag{4}$$

$$N(x) \ge N(y), \quad \text{if } x \le y, \qquad \textit{(antimonotonicity)} \tag{5}$$

$$N(N(x)) =x. \tag{6}$$

Generally fuzzy negation is defined by (4)-(5). Most popular fuzzy conjunction and disjunction functions are called t-norms and t-conorms [8] and defined by commutativity, associativity and (7)-(9) axioms:

$$T(x,1) = x, \qquad T(1,y) = y, \qquad \textit{(boundary conditions)} \tag{7}$$

$$S(x,0) = x, \qquad S(0,y) = y, \qquad \textit{(boundary conditions)} \tag{8}$$

$$T(x,y) \le T(u,v), \quad S(x,y) \le S(u,v), \quad \textit{if } x \le u,\ y \le v. \quad \textit{(monotonicity)} \tag{9}$$

As it was pointed out in [4-6] associativity and commutativity are very useful properties of these operations in mathematical models of fuzzy logic but in industrial applications of fuzzy systems these properties do not play important role. For this reason fuzzy conjunction and disjunction can be defined only by axioms (7)-(9). Another reason for such definition of these operations is that due to associativity parametric t-norms and t-conorms usually have complicated forms for tuning in fuzzy systems and for hardware implementation.

From (7)-(9) it follows:

$$T(x,0) = 0, \ \ T(0,y) = 0, \tag{10}$$

$$S(x,1) = 1, \ \ S(1,y) = 1, \tag{11}$$

and, hence, the properties (2)-(3) are fulfilled in points {0,1} for fuzzy conjunctions and disjunctions defined by (7)-(9).

Consider the following three pairs of simplest t-norms and t-conorms:

$$T_M(x,y) = min\{x,y\}, \quad (minimum), \qquad S_M(x,y) = max\{x,y\}, \quad (maximum), \tag{12}$$

$$T_L(x,y)= max\{x+y-1, 0\}, \quad S_L(x,y)= min\{x+y, 1\}, \qquad (Lukasiewicz), \tag{13}$$

$$T_D(x,y) = \begin{cases} 0, & if \ (x,y) \in [0,1)\times[0,1) \\ \min(x,y), & otherwise \end{cases}, \qquad (drastic\ product), \tag{14}$$

$$S_D(x,y) = \begin{cases} 1, & if \ (x,y) \in (0,1]\times(0,1] \\ \max(x,y), & otherwise \end{cases} \qquad (drastic\ sum). \tag{15}$$

These three pairs of conjunction and disjunction operations will be considered as basic conjunction and disjunction operations. These operations have efficient hardware implementation [7] and they will be used further for generation of new parametric classes of conjunction and disjunction operations. Note that the product $t$-norm $T_P(x,y) = x\cdot y$, and the probabilistic sum $S_P(x,y) = x + y - x\cdot y$ have less effective digital hardware implementation than basic operations (12)-(15) due to presence of product operation in their definition [7].

Pairs of $t$-norms and $t$-conorms considered above can be obtained one from another by means of negation $N(x)= 1 - x$ as follows:

$$S(x,y) = N(T(N(x),N(y))), \qquad T(x,y) = N(S(N(x),N(y))). \tag{16}$$

If $N$ is an involution then these relations also can be used for obtaining from a given fuzzy conjunction $T$ (or disjunction $S$) corresponding disjunction S (conjunction $T$). In this case the obtained pair of fuzzy conjunction and disjunction operations together with involution $N$ will constitute a De Morgan triple *(N,T,S)*. Further we will consider the methods of generation of fuzzy conjunctions. Fuzzy disjunctions can be obtained from them by means of (16).

It can be shown [5] that fuzzy conjunctions and disjunctions satisfy the following inequalities:

$$T_D(x,y) \le T(x,y) \le T_M(x,y) \le S_M(x,y) \le S(x,y) \le S_D(x,y), \tag{17}$$

so, min, max and drastic operations define the borders for fuzzy conjunctions and disjunctions.

Several methods of generation of simple parametric fuzzy operations suitable for tuning in fuzzy systems have been proposed in [4-6]. These methods are based on the formula:

$$T(x,y)= T_2(T_1(x,y),S(x,y)), \tag{18}$$

where $T_2$ and $T_1$ are conjunctions and $S$ is a pseudo-disjunction. Recall that *pseudo-conjunctions T* and *pseudo-disjunctions S* are defined in [5] as functions satisfying (9)-(11), respectively. Suppose $h,g_1,g_2:L\rightarrow L$ are non-decreasing functions called generators. Pseudo-disjunction $S$ can de generated as follows:

$$S(x,y)= S_1(g_1(x),g_2(y)), \tag{19}$$

$$S(x,y)= g_1(S_1(x,y)), \tag{20}$$

$$S(x,y)= S_2(S_1(x,y),h(y)), \tag{21}$$

where $g_1$ and $g_2$ satisfy conditions: $g_1(1)= g_2(1)= 1$. As pseudo-disjunctions $S$ in (18) one can use basic disjunctions or pseudo-disjunctions obtained from basic disjunctions by recursive application of (18)-(21). Several classes of parametric conjunctions simple for tuning in fuzzy systems were proposed in [4-6] but most of them have not efficient hardware implementation because they are based on product operation or exponential function.

In the present paper we study the problem of construction of wide class of fuzzy parametric conjunction and disjunction operations with effective hardware implementation. We propose several classes of generators simple for hardware implementation. These generators together with basic conjunctions and disjunctions can be used to generate parametric classes of fuzzy conjunctions with effective hardware implementation.

## 3 Digital Generators of Fuzzy Operations

Suppose it is used $m$ bits in digital representation of membership values. Then different membership values can be represented by $2^m$ numbers from the set $L_D= \{0,1,2,\ldots, 2^m-1\}$. Denote the maximal membership value $I = 2^m-1$. This value will represent the full membership corresponding to the value 1 in a traditional set of membership values $L= [0,1]$. All definitions and properties of fuzzy operations (1)-(21) from the previous section can be transformed into the digital case by replacing the set of membership values $L= [0,1]$ by $L_D= \{0,1,2,\ldots, 2^m-1\}$ and maximal membership value 1 by I. Also in definition of drastic product and sum in (14), (15) we need to replace intervals [0,1) and (0,1] by the sets $\{0,1,2, \ldots ,2^m-2\}$ and $\{1,2,\ldots, 2^m-1\}$, respectively. For graphical representation of digital generators and fuzzy operations we will use below $m= 4$ bits, with the following set of digital membership values $L_D=\{0,1,2,\ldots,14,15\}$ and maximal membership value I= 15.

We need to use in (19)-(21) generators that have effective hardware implementation. For this reason we propose the simplest generators depending on one parameter $p$. We suppose that parameter $p$ can change from 0 till I. Suppose 0.5(-) and 0.5(+) denote digital representations of the membership value 0.5. They are defined as follows: 0.5(-) = $2^{m-1}$-1, 0.5(+) = $2^{m-1}$. For example, for $m$= 4 bits we have I = $2^m$-1= 15, 0.5(-) = 7 and 0.5(+) = 8. We propose the following simplest generators that have efficient hardware implementation:

$$g(x) = \begin{cases} 0, & if \quad x < p \\ I, & otherwise \end{cases}, \qquad (0\text{I}\_step) \tag{22}$$

$$g(x) = \begin{cases} 0, & if \quad x \le p \\ x, & otherwise \end{cases} \qquad (0\text{-}diag) \tag{23}$$

$$g(x) = \begin{cases} x, & if \quad x \le p \\ I, & otherwise \end{cases} \qquad (\text{I-}diag) \tag{24}$$

$$g(x) = \begin{cases} 0, & if \quad x \le p \\ x, & if \quad p < x < I - p \\ I, & otherwise \end{cases} \qquad (0I\text{-}diag) \tag{25}$$

$$g(x) = \begin{cases} x, & if \quad x \le p \\ p, & if \quad p < x \le 0.5(-) \\ I - p, & if \quad 0.5(+) \le x < I - p \\ I, & otherwise \end{cases} \qquad (p\text{-}diag) \tag{26}$$

$$g(x) = \begin{cases} \max(x - p, 0), & if \quad x \le 0.5(-) \\ \min(x + p, I), & otherwise \end{cases} \qquad (bound\text{-}dif\text{-}sum) \tag{27}$$

Fig.1. shows these generators in digital representation with 4 bits for parameter value $p$ = 3 corresponding to value 0.2 in interval [0,1] of true values.

Fig.2 contains an example of parametric fuzzy conjunction obtained by (18) and (19) by means of basic fuzzy conjunctions and disjunction operations: $T_1$= minimum, $T_2$= Lukasiewicz conjunction, $S_1$= maximum, and with generators: $g_1$= $p$-diag, $g_2$= 0I-step. The value of the parameters $p$ in generator $p$-diag is equal to 0.2 in [0,1] scale of true values. For digital representation with 4 bits and I = 15 it corresponds to value $p$= 3. Parameter $p$ in generator $g_2$= 0I-step equals to 0.7 in [0,1] scale or 11 in 4 bits representation.

Fig. 3 contains an example of parametric fuzzy conjunction obtained by (18) and (20) by means of generator 0I-diag and basic fuzzy conjunctions and disjunction

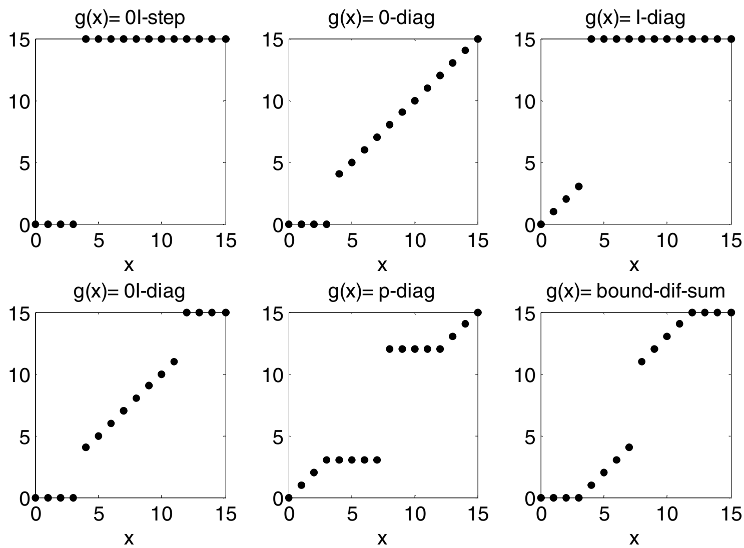


**Fig. 1.** Generators with parameter $p$= 3 in digital representation with 4 bits

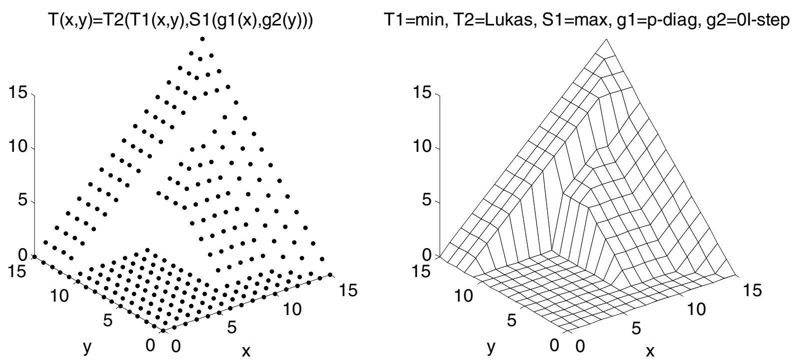


**Fig. 2.** Example of parametric fuzzy conjunction obtained by (18) and (19). On the left it is shown a shape of conjunction function with digital representation with 4 bits and on the right it is presented the same shape as a mesh figure.

operations: $T_1$= Lukasiewicz conjunction, $T_2$= drastic conjunction, $S$= maximum. The value of the parameter $p$ in generator 0I-diag is equal to 0.2 in [0,1] scale of true values or 3 in 4 bits representation.

Fig. 4 contains an example of parametric fuzzy conjunction obtained by (18) and (21) by means of generator $h$= bound-dif-sum and basic fuzzy conjunctions and disjunction operations: $T_1$= minimum, $T_2$= Lukasiewicz conjunction, $S_1$= maximum, $S_2$= drastic disjunction. The shape of obtained fuzzy conjunction corresponds to the value of the parameter $p$ in generator bound-dif-sum equal to 0.4 in [0,1] scale of true values and 6 in digital representation with 4 bits.

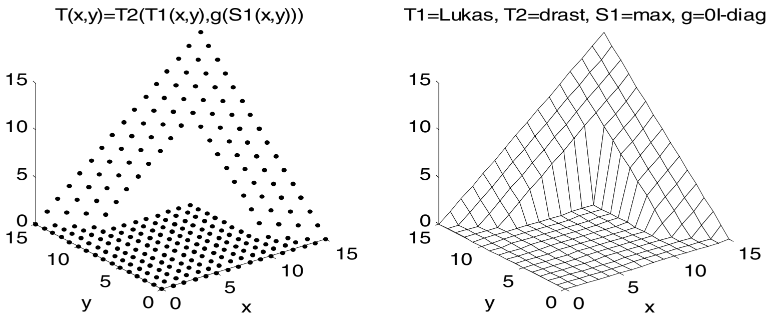


**Fig. 3.** Example of parametric fuzzy conjunction obtained by (18) and (20)

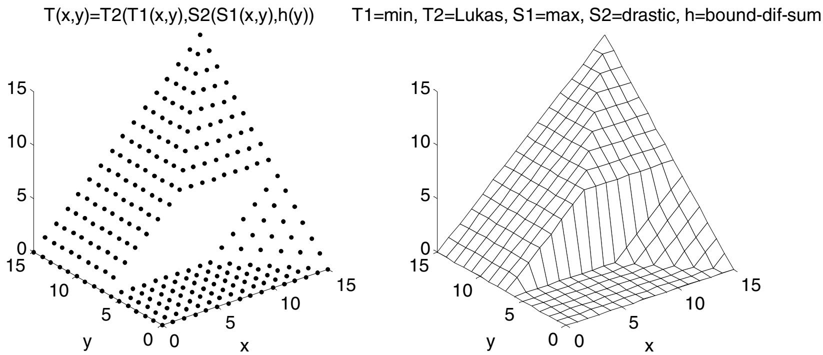


**Fig. 4.** Example of parametric fuzzy conjunction obtained by (18) and (21)

## 4 Digital Hardware Implementation of Parametric Fuzzy Operations

In this section we introduce basic functions that can be used for generation and hardware implementation of generators and basic fuzzy conjunctions and disjunctions considered in previous sections. To apply (18)-(21) we need to use functions that have

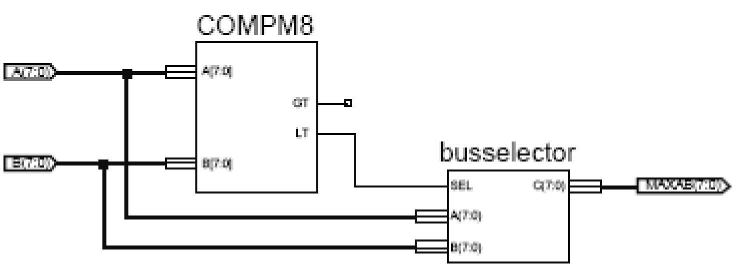


**Fig. 5.** Hardware implementation of maximum operation

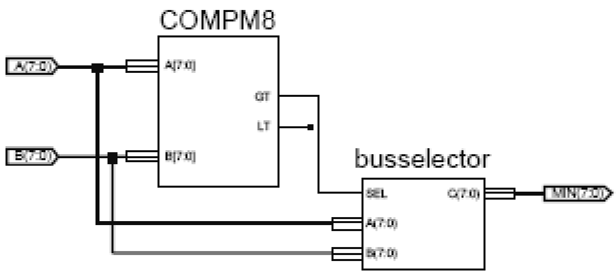


**Fig. 6.** Hardware implementation of minimum operation

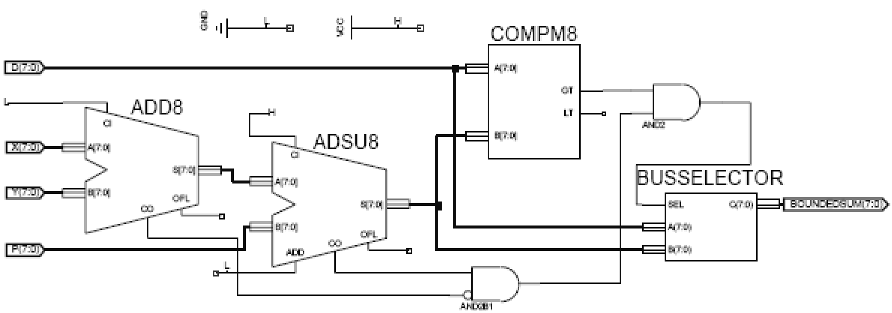


**Fig. 7.** Hardware implementation of bounded sum operation

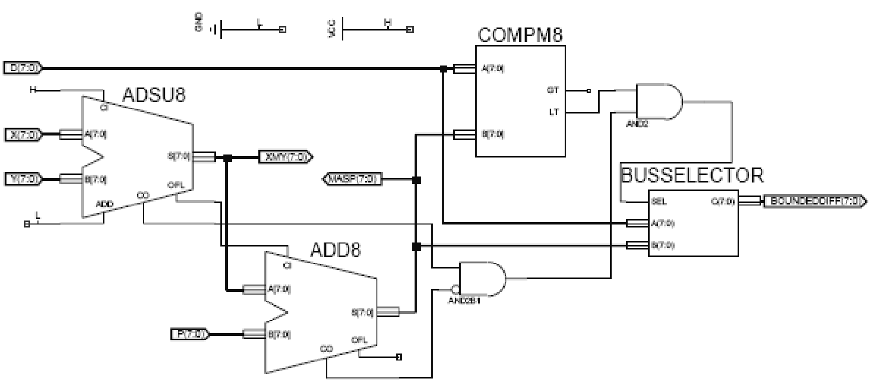


**Fig. 8.** Hardware implementation of bounded difference operation

effective digital hardware implementation. These functions we can obtain as a superposition of the following *basic functions* which have effective digital hardware implementation: constant, identity, minimum, maximum, bounded sum: $F(x,y)= min(\mathrm{I},x+y)$, and bounded difference: $F(x,y)= max(0,x-y)$. Also comparison operations can be used as basic functions in superposition of considered above functions. It is clear that the basic conjunctions and disjunctions (12)-(15) and introduced generators (22)-(27) can be considered as compositions of these basic functions. The circuits corresponding to basic functions are realized using Xilinx tools for FPGA design.

Basic logical gates, comparator, adder and subtractor [14-16] are used to construct the circuits. The circuit of 8-bits digital hardware implementation of the main basic functions are shown in Fig. 5-8. These circuits can be used as bricks in construction of generators and fuzzy conjunction and disjunction operations considered in this paper.

## 5 Conclusions

The main contribution of the paper is the following. The problem of effective digital hardware implementation of fuzzy systems with parametric conjunction and disjunction operations is formulated and its solution is outlined. Most of known parametric fuzzy conjunction and disjunction operations have not effective hardware implementation because they use product, division, exponentiation or other operations that have not effective hardware implementation. For example, minimum and maximum are simple comparisons between two values, which on digital circuitry is easy to implement, but a product is much more complex because of the following reasons [15,16]:

1. It requires at least $n$-1 iterations on a sequential multiplier with $n$ bits.
2. For the case of a combinatorial array circuit $n$-1 levels of adders are required for each pair of $n$ bits used for input length.
3. Shifting operation can be realized minimizing time and resources but only for multiplying a number by a power of two.

A proposed new approach is based on the use of simple generators that can be composed with simple fuzzy conjunction and disjunction operations. Several classes of such parametric generators are proposed and examples of parametric fuzzy conjunctions obtained by means of these generators are presented. Further, the basic functions that can be used in construction of generators and basic conjunction and disjunction functions are considered. The circuits defining digital hardware implementation of these functions are presented. These circuits can be used as bricks in hardware implementation of generators and fuzzy conjunctions and disjunctions. The obtained results can be extended in several directions. First, the proposed approach can be extended on generation of parametric t-norms and t-conorms that have efficient digital hardware implementation. Second, obtained results can be used in digital hardware implementation of inference and aggregation operations in fuzzy systems with parametric conjunctions and disjunctions. Hardware implementation of such systems will extend possibilities of design of flexible on-board and real-time fuzzy systems that can be used as components of applied intelligent systems in control, pattern recognition and decision making.

**Acknowledgments.** The research work was partially supported by CONACYT, NKTH, OTKA, SIP – IPN, Project No. 20070945.

## References

1. Terano, T., Asai, K., Sugeno, M.: Applied Fuzzy Systems. Academic Press Professional, San Diego (1994)
2. Yen, J., Langari, R., Zadeh, L.A.: Industrial Applications of Fuzzy Logic and Intelligent Systems. IEEE Press, NJ (1995)

3. Kandel, A., Langholz, G.: Fuzzy Hardware: Architectures and Applications. Kluwer Academic Publishers, Dordrecht (1997)
4. Batyrshin, I., Kaynak, O., Rudas, I.: Generalized Conjunction and Disjunction Operations for Fuzzy Control. In: EUFIT 1998, 6th European Congress on Intelligent Techniques & Soft Computing, vol. 1, pp. 52–57. Verlag Mainz, Aachen (1998)
5. Batyrshin, I., Kaynak, O.: Parametric Classes of Generalized Conjunction and Disjunction Operations for Fuzzy Modeling. IEEE Transactions on Fuzzy Systems 7, 586–596 (1999)
6. Batyrshin, I., Kaynak, O., Rudas, I.: Fuzzy Modeling Based on Generalized Conjunction Operations. IEEE Transactions on Fuzzy Systems 10, 678–683 (2002)
7. Batyrshin, I., Hernández Zavala, A., Camacho Nieto, O., Villa Vargas, L.: Generalized Fuzzy Operations for Digital Hardware Implementation. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 9–18. Springer, Heidelberg (2007)
8. Klement, E.P., Mesiar, R., Pap, E.: Triangular Norms. Kluwer, Dordrecht (2000)
9. Jang, J.-S.R., Sun, C.T., Mizutani, E.: Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence. Prentice-Hall International, Englewood Cliffs (1997)
10. Fodor, J.: A New Look at Fuzzy Connectives. Fuzzy Sets Syst. 57, 141–148 (1993)
11. Batyrshin, I.: On the Structure of Involutive, Contracting and Expanding Negations. Fuzzy Sets and Syst. 139, 661–672 (2003)
12. Klir, G.J., Folger, T.A.: Fuzzy Sets, Uncertainty, and Information. Prentice-Hall, NJ (1988)
13. De Cooman, G., Kerre, E.E.: Order Norms on Bounded Partially Ordered Sets. J. Fuzzy Math. 2, 281–310 (1994)
14. Tocci, R.J., Widmer, N.S., Moss, G.L.: Digital Systems: Principles and Applications, 9th edn. Prentice Hall, Englewood Cliffs (2003)
15. Patterson, D.A., Hennessy, J.L.: Computer Organization and Design: The Hardware / Software Interface, 2nd edn. Morgan Kaufmann, San Francisco (1998)
16. Zargham, M.R.: Computer Architecture: Single and Parallel Systems. Prentice-Hall, Englewood Cliffs (1995)