Generalized Fuzzy Operations for Digital Hardware Implementation

Ildar Batyrshin[1], Antonio Hernández Zavala[2], Oscar Camacho Nieto[2], and Luis Villa Vargas[2]

[1] Mexican Petroleum Institute
batyr1@gmail.com
[2] Mexican Polytechnic Institute – Computer Research Centre
antonioh@hotmail.com, oscarc@cic.ipn.mx, lvilla@cic.ipn.mx

Abstract. Hardware implementation of fuzzy systems plays important role in many industrial applications of fuzzy logic. The most popular applications of fuzzy hardware systems were found in the domain of control systems but the area of application of these systems is extending on other areas such as signal processing, pattern recognition, expert systems etc. The digital fuzzy hardware systems usually use only basic operations of fuzzy logic like min, max and some others, first, due to their popularity in traditional fuzzy control systems and, second, due to the difficulties of hardware implementation of more complicated operations, e.g. parametric classes of *t*-norms and *t*-conorms. But for extending the area of applications and flexibility of fuzzy hardware systems it is necessary to develop the methods of digital hardware implementation of wide range of fuzzy operations. The paper studies the problem of digital hardware implementation of fuzzy parametric conjunction and disjunction operations. A new class of such operations is proposed which is simple for digital hardware implementation and is flexible, due to its parametric form, for possible tuning in fuzzy models. The methods of hardware implementation of these operations in digital systems are proposed.

Keywords: Fuzzy system, conjunction, disjunction, digital system.

1 Introduction

Fuzzy logic gives possibility to translate human knowledge into rules of fuzzy systems. Such systems have wide applications often providing better results than conventional techniques [1-3]. For on-board and real-time applications the fuzzy systems require faster processing speed which makes specific inference hardware the choice to satisfy processing time demands. Hardware implementation of fuzzy systems plays an important role in many industrial applications of fuzzy logic, mainly in the domain of control systems, but the area of application of these systems is extending on other areas such as signal processing, pattern recognition, expert systems etc [4 -6]. Departure point for digital fuzzy processors was at mid 80's by Togai and Watanabe [7] and, since then, many architectures have been presented with faster inferences and decreasing hardware complexity [6, 8-11]. FPGA technology

A. Gelbukh and A.F. Kuri Morales (Eds.): MICAI 2007, LNAI 4827, pp. 9–18, 2007.

has become a fast design alternative to implement complex digital systems, given its reprogrammable capabilities. This technology have been used by some researchers to adapt the mathematics of fuzzy logic systems into digital circuitry [12-15].

The digital fuzzy hardware systems usually use as fuzzy conjunction and disjunction only basic operations of fuzzy logic like *min, max* and product [16], first, due to their popularity in traditional fuzzy control systems and, second, due to the difficulties of hardware implementation of more complicated operations, e.g. parametric classes of *t*-norms and *t*-conorms [17]. Among mentioned operations, *min* and *max* are simple comparisons between two values, which on digital circuitry is easy to implement; product is much more complex, because of the following reasons [18,19]:

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.

Fuzzy rule based systems are usually constructed based on expert knowledge and on experimental data describing system behavior. First fuzzy systems usually have been based on expert knowledge and trial-and-error adjustment of rules and membership functions. At the beginning of 90's it was proved that fuzzy systems are universal approximators [20,21]. This fundamental result in fuzzy set theory stimulated development of different methods of fuzzy systems optimization based on automatic adjustment of rules and membership functions [3, 22, 23]. Another approach to fuzzy system optimization was proposed in [24, 25] where instead of adjusting of membership functions it was proposed to adjust parameters of fuzzy operations used in fuzzy systems. This approach gives possibility to keep unchanged expert knowledge about fuzzy concepts given in membership functions. Instead of traditional *t*-norms and *t*-conorms sufficiently complicated for tuning in optimization process it was proposed to introduce simple parametric fuzzy conjunction and disjunction operations satisfying simplified system of axioms [24,25] in contrast to *t*-norms and *t*-conorms satisfying very restrictive associativity property [17].

In this paper it is considered a problem of hardware implementation of fuzzy systems obtained as a result of adjusting of parametric conjunction and disjunction operations. It should be noted that both parametric *t*-norms [17] and parametric generalized conjunctions considered in [24, 25] use the product operation as a constituent. As it was mentioned above the product operation has not sufficiently efficient hardware implementation. To avoid this problem this paper introduces new parametric family of fuzzy conjunction operations without product operation as a constituent. Disjunction operations can be obtained dually to conjunctions operations.

The paper has the following structure. Section 2 gives the basic definitions of fuzzy conjunction and disjunction operations. In Section 3 a new method of generation of non-associative conjunctions suitable for hardware implementation is proposed. The methods of a hardware implementation of proposed parametric conjunction operations are considered is Section 4. In Conclusion we discuss obtained results and future directions of research.

2 Basic Definitions

Triangular norm (t-norm) T and triangular conorm (t-conorm) S are defined as functions T,S:[0,1]×[0,1]→[0,1] satisfying on [0,1] the following axioms [17]:

$$T(x,y) = T(y,x), \qquad S(x,y) = S(y,x), \qquad (commutativity)$$

$$T(T(x,y),z) = T(x,T(y,z)), \qquad S(S(x,y),z) = S(x,S(y,z)), \qquad (accociativity)$$

$$T(x,y) \le T(u,v), \qquad S(x,y) \le S(u,v), \quad \text{if } x \le u,\ y \le v \qquad (monotonicity)$$

$$T(x,1) = x, \qquad S(x,0) = x \qquad (boundary\ conditions)$$

From this definition the following properties are followed:

$$T(0,x)= T(x,0)= 0,\ S(1,x)= S(x,1)= 1,\ T(1,x)= T(x,1)= x,\ S(0,x)= S(x,0)= x \qquad (1)$$

t-norm and t-conorm are dual.

An involutive negation is a function N:[0,1]→[0,1] satisfying on [0,1] the following conditions:

$$n(x) \le n(y) \qquad \text{if } y \le x,$$

$$n(0) = 1, \qquad n(1) = 0,$$

$$n(n(x)) = x.$$

t-norm and t-conorm can be obtained one from another by means of negation operation as follows:

$$S(x,y) = n(T(n(x),n(y))), \qquad T(x,y) = n(S(n(x),n(y))).$$

The following are the simplest t-norm and t-conorm mutually related by De Morgan lows with negation operation $n(x) = 1 - x$:

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

$$T_P(x,y) = x\cdot y \quad (product), \qquad S_P(x,y) = x + y - x\cdot y, \qquad (probabilistic\ sum),$$

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

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

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

Lukasiewicz t-norm T_L and t-conorm S_L are also called a bounded product and bounded sum, respectively.

All *t*-norms *T* and *t*-conorms *S* satisfy on [0,1] the following inequalities:

$$T_D(x,y) \le T(x,y) \le T_M(x,y) \le S_M(x,y) \le S(x,y) \le S_D(x,y). \tag{2}$$

As follows from these inequalities $T_D(x,y)$, $T_M(x,y)$, $S_M(x,y)$ and $S_D(x,y)$ serve as boundaries for all *t*-norms and *t*-conorms.

Several families of parametric *t*-norms and *t*-conorms can be found in [17]. Below is an example of Dombi *t*-norm, depending on parameter $\lambda \in [0, \infty]$:

$$T(x,y) = \frac{1}{1+\left(\left(\frac{1-x}{x}\right)^{\lambda}+\left(\frac{1-y}{y}\right)^{\lambda}\right)^{\frac{1}{\lambda}}} \qquad \text{if } \lambda \in (0,\infty),$$

$$T(x,y) = T_D(x,y), \qquad \text{if } \lambda = 0,$$

$$T(x,y) = T_M(x,y), \qquad \text{if } \lambda = \infty.$$

As it was noted in [24], parametric *t*-norms and *t*-conorms have sufficiently complicated form due to the associativity property requiring to use inverse functions of generators of these operations [17]. For this reason, traditional parametric t-norms are sufficiently complicated for automatic adjusting of their parameters in automatic optimization of fuzzy systems. To obtain more simple parametric *t*-norms in [24,25] it was proposed to use non-associative conjunction operations. Usually the property of associativity does not used in construction of applied fuzzy systems where position of operands of these operations is fixed. Moreover, often only two operands are used in these operations as in fuzzy control systems with two input variables. Several methods of parametric non-associative conjunctions were proposed in [24]. One of such methods is following:

$$T(x,y) = T_2(T_1(x,y), S(g_1(x), g_2(y)))$$

where T_1 and T_2 are some conjunctions, S is a disjunction, and g_1, g_2 are non-decreasing functions g_1, g_2:[0,1]→[0,1] such that $g_1(1)= g_2(1)= 1$. As T_1 and T_2 it can be used for example one of basic *t*-norms considered above. Some examples of simple parametric conjunctions obtained by this method are following:

$$T(x,y) = \begin{cases} \min(x,y), & if \quad p \le x \ or \ q \le y \\ 0, & otherwise \end{cases}, \tag{3}$$

$$T(x,y) = min(x,y) \cdot max\{1 - p(1-x), 1 - q(1-y), 0\}, \tag{4}$$

$$T(x,y) = min\{min(x,y), max(x^p, y^q)\}, \tag{5}$$

$$T(x,y) = min(x,y) \cdot max(x^p, y^q).$$

Note that inequality (2) is also fulfilled for non-associative conjunctions.

In [25] more generalized conjunction operations defined by monotonicity property and simplified boundary conditions:

$$T(0,0) = T(0,1) = T(1,0) = 0, \qquad T(1,1) = 1,$$

were considered. The following methods of generation of such operations were proposed:

$$T(x,y) = T_2(T_1(x,y), S_1(g_1(x),g_2(y))),$$

$$T(x,y) = T_2(T_1(x,y), g_1(S_1(x,y))),$$

$$T(x,y) = T_2(T_1(x,y), S_2(h(x),S_1(x,y))),$$

where S is a monotone function satisfying conditions:

$$S(1,0)= S(0,1)= S(1,1)= 1$$

and g_1, g_2, h are non-decreasing functions $g_1, g_2, h{:}[0,1]\to[0,1]$ such that $g_1(1)= g_2(1)= h(1)= 1$. These methods gives possibility to generate the following simplest conjunction operations :

$$T(x,y) = min(x^p,y^q),$$

$$T(x,y) =x^p y^q,$$

$$T(x,y) =(xy)^p(x + y - xy)^q.$$

Parametric conjunctions introduced in [24,25] are simpler than most of known parametric *t*-norms and suitable for their adjusting in optimization of fuzzy models but hardware implementation of most of these operations is still non-effective due to the presence of operations product and computing powers in their definitions. For this reason only parametric operation (3) considered above can have effective hardware implementation. In the following section we propose a new method of generation of non-associative conjunctions which can give possibility to construct parametric conjunctions based only on basic *min, max,* Lukasiewicz and drastic operations which have effective hardware implementation.

3 New Method of Generation of Non-associative Conjunctions

We propose the following new method of generation of conjunctions:

$$T(x,y) = min(T_1(x,y),S(T_2(x,y),s)), \tag{6}$$

where T_1 and T_2 are some conjunctions, S is a disjunction, and s is a parameter $s\in[0,1]$.

The following properties of (6) can be proved.

Theorem 1. If T_1, T_2 and S are commutative, monotonic functions satisfying boundary conditions (1) then T is the same.

Proposition 2. For specific *t*-norms and *t*-corms (6) is reduced as follows:

$$\text{if } T_2 = T_M \text{ then } T(x,y) = T_1(x,y);$$

$$\text{if } T_1 = T_D \text{ then } T(x,y) = T_D(x,y);$$

$$\text{if } T_1 = T_2 \text{ then } T(x,y) = T_1(x,y).$$

Based on (6) and avoiding cases considered in Proposition 2 by means of basic *t*-norms and *t*-conorms T_M, T_L and T_D we can obtain new parametric conjunctions with effective hardware implementation. For example we can introduce the following simple parametric conjunctions:

$$T(x,y) = min(min(x,y), S_L(T_L(x,y),s)), \quad (7)$$

$$T(x,y) = min(min(x,y), S_L(T_{MB}(x,y),s)), \quad (8)$$

where T_{MB} denotes a conjunction (3). Fig. 1 depicts the shape of conjunction (7) and Fig. 2 depicts the shape of conjunction (8). In these pictures parameters s,p,q define the sizes of the "holes" in the "pyramid" corresponding to the conjunction T_M.

In the following section we consider examples of hardware implementation of some new parametric conjunctions.

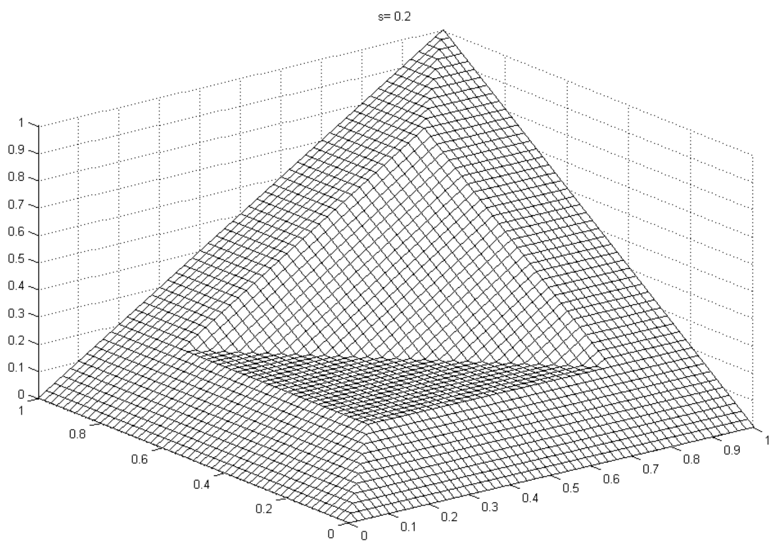

Fig. 1. The shape of the conjunction (7)

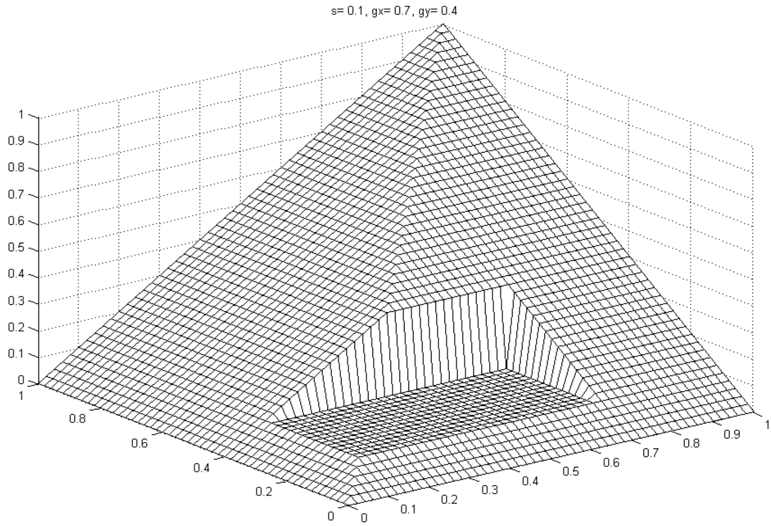

Fig. 2. The shape of the conjunction (8)

4 Hardware Implementation of Fuzzy Conjunctions

Fuzzy membership value is commonly expressed as a floating point number in interval [0, 1], where 0 means non membership and 1 means complete membership and generally can have infinite values. Unfortunately this is not so easy for a computer to do calculations by using this representation, instead of this we can work with integers between [0, 2^n-1], where n is the number of bits used to represent truth space and gives the resolution, 0 means no membership, and 2^n-1 is the complete membership value. We consider here 8 bit resolution of numbers. Denote $d = 2^n-1$, n=8.

Below are the methods of hardware implementation of Lukasiewicz t-norm and t-conorm (bounded product and bounded sum) are discussed. These circuits were realized using Xilinx tools for FPGA design, obtaining equivalent circuits for mentioned operations. Basic logical gates [26] are used to construct the circuits, some operations used are common constructs on digital hardware this is the case of comparator, adder and subtractor [18,19]. The hardware implementation of bounded product is shown in Fig. 3.

In the first block from left, there is an adder to implement $x+y$, if the sum of this values is more than 8 bit value count there is a carry output flag to indicate that data is not valid. Second block subtract d from result of previous block, there is also a carry output flag to identify when data is not valid. In order to obtain a valid data up to this part it is necessary to have two valid conditions on two previous blocks, this correspond to CO=0 on the addition block and CO=1 on the subtraction block, this is realized by an AND gate. Third and fourth blocks correspond to the maximum operation between 0 and valid result from previous stage, AND gate between both blocks is on charge to decide if there is no valid data, let 0 be the output value.

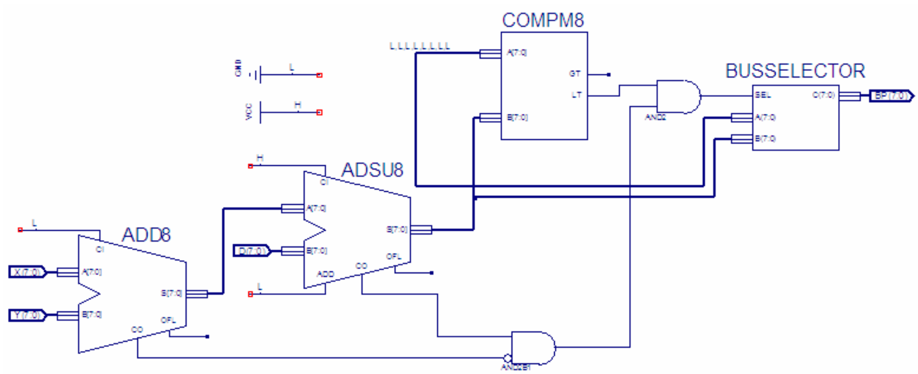

Fig. 3. Digital hardware for bounded product

Circuit shown in Fig.4 corresponds to a bounded sum operation. Its functioning is alike previous circuit in two first blocks, the main difference here is that last two blocks perform minimum operation.

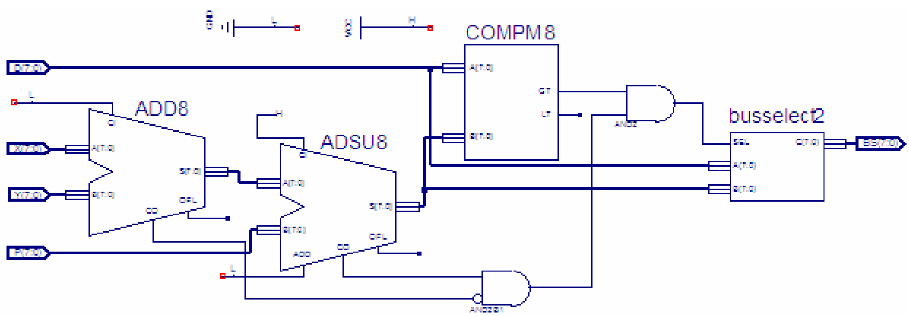

Fig. 4. Digital hardware for bounded sum operation

A new parametric conjunction (7) is represented in Fig. 5. Here, each block corresponds to the circuit diagrams shown before.

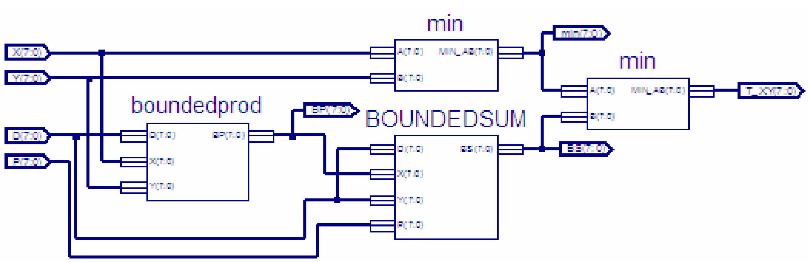

Fig. 5. Hardware implementation of parametric conjunction (7)

5 Conclusions

The main contribution of the paper is the following. The problem of effective digital hardware implementation of fuzzy parametric conjunction and disjunction operations is formulated and its solution is proposed. It is a first step in hardware implementation of fuzzy systems obtained as a result of adjusting of parameters of operations. Most of known parametric fuzzy conjunction and disjunction operations have not effective hardware implementation because they use product and computing powers operations in their definitions. New method of generation of parametric conjunction and disjunction operations is proposed in this paper. Based on this method new parametric classes of fuzzy operations, suitable for effective hardware implementation, are obtained. The methods of hardware implementation of some of these operations are given. The obtained results can be extended in several directions. First, effective hardware implementation of parametric operations based on drastic *t*-norm and *t*-corm can be done in the similar manner as it was done for operations based on Lukasiewicz operations. 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.

Acknowledgments. The research work was partially supported by CONACYT and 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. Jang, J.-S.R., Sun, C.T., Mizutani, E.: Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence (1997)
4. Jespers, P.G.A., Dualibe, C., Verleysen, M.: Design of Analog Fuzzy Logic Controllers in CMOS Technologies. In: Implementation, Test and Application, Kluwer Academic Publishers, New York (2003)
5. Kandel, A., Langholz, G.: Fuzzy Hardware: Architectures and Applications. Kluwer Academic Publishers, Dordrecht (1997)
6. Patyra, M.J., Grantner, J.L., Koster, K.: Digital Fuzzy Logic Controller: Design and Implementation. IEEE Transactions on Fuzzy Systems 4, 439–459 (1996)
7. Togai, M., Watanabe, H.: A VLSI Implementation of a Fuzzy-Inference Engine: Toward an Expert System on a Chip. Information Sci. 38, 147–163 (1986)
8. Cardarilli, G.C., Re, M., Lojacono, R., Salmeri, M.: A New Architecture for High-Speed COG Based Defuzzification. In: TOOLMET 1997. International Workshop on Tool Environments and Development Methods for Intelligent Systems, pp. 165–172 (1997)
9. Gaona, A., Olea, D., Melgarejo, M.: Distributed Arithmetic in the Design of High Speed Hardware Fuzzy Inference Systems. In: 22nd International Conference of the North American Fuzzy Information Processing Society, pp. 116–120 (2003)

10. Banaiyan, A., Fakhraie, S.M., Mahdiani, H.R.: Cost-Performance Co-Analysis in VLSI Implementation of Existing and New Defuzzification Methods. In: Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, vol. 1, pp. 828–833 (2005)
11. Tamukoh, H., Horio, K., Yamakawa, T.: A Bit-Shifting-Based Fuzzy Inference for Self-Organizing Relationship (SOR) Network. IEICE Electronics Express 4, 60–65 (2007)
12. Salapura, V., Hamann, V.: Implementing Fuzzy Control Systems Using VHDL and Statecharts. In: EURO-DAC 1996 with EURO-VHDL 1996. Proc. of the European Design Automation Conference, pp. 53–58. IEEE Computer Society Press, Geneva (1996)
13. Sánchez-Solano, S., Senhadji, R., Cabrera, A., Baturone, I., Jiménez, C.J., Barriga, A.: Prototyping of Fuzzy Logic–Based Controllers Using Standard FPGA Development Boards. In: Proc. IEEE International Workshop on Rapid System Prototyping, pp. 25–32. IEEE Computer Society Press, Los Alamitos (2002)
14. Garrigos-Guerrero, F.J., Ruiz Merino, R.: Implementacion de Sistemas Fuzzy Complejos sobre FPGAs. In: Computación Reconfigurable y FPGAs, pp. 351–358 (2003)
15. Raychev, R., Mtibaa, A., Abid, M.: VHDL Modelling of a Fuzzy Co-Processor Architecture. In: International Conference on Computer Systems and Technologies – CompSysTech (2005)
16. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)
17. Klement, E.P., Mesiar, R., Pap, E.: Triangular Norms. Kluwer, Dordrecht (2000)
18. Patterson, D.A., Hennessy, J.L.: Computer Organization and Design: The Hardware/Software Interface, 2nd edn. Morgan Kaufmann Publishers, San Francisco (1998)
19. Zargham, M.R.: Computer Architecture: Single and Parallel Systems. Prentice-Hall, Englewood Cliffs (1995)
20. Kosko, B.: Fuzzy Systems as Universal Approximators. In: Proc. IEEE Int. Conf. on Fuzzy Systems, pp. 1153–1162. IEEE Computer Society Press, Los Alamitos (1992)
21. Wang, L.X.: Fuzzy Systems are Universal Approximators. In: Proc. IEEE Int. Conf. On Fuzzy Systems, pp. 1163–1170. IEEE Computer Society Press, Los Alamitos (1992)
22. Kosko, B.: Fuzzy Engineering. Prentice-Hall, New Jersey (1997)
23. Wang, L.-X.: A Course in Fuzzy Systems and Control. Prentice Hall PTR, Upper Saddle River, NJ (1997)
24. Batyrshin, I., Kaynak, O.: Parametric Classes of Generalized Conjunction and Disjunction Operations for Fuzzy Modeling. IEEE Transactions on Fuzzy Systems 7, 586–596 (1999)
25. Batyrshin, I., Kaynak, O., Rudas, I.: Fuzzy Modeling Based on Generalized Conjunction Operations. IEEE Transactions on Fuzzy Systems 10, 678–683 (2002)
26. Tocci, R.J., Widmer, N.S., Moss, G.L.: Digital Systems: Principles and Applications, 9th edn. Prentice-Hall, Englewood Cliffs (2003)