Digital Fuzzy Parametric Conjunctions for Hardware Implementation of Fuzzy Systems


* Budapest Tech, Budapest, Hungary
** Mexican Petroleum Institute, D.F., Mexico
*** Mexican Polytechnic Institute/Computer Research Centre, D.F., Mexico

rudas@bmf.hu, batyr1@gmail.com, antonioh@hotmail.com, {oscarc, lvilla}@cic.ipn.mx

Abstract—The problem of construction of parametric classes of fuzzy conjunction and disjunction operations suitable for efficient digital hardware implementation is studied. The set of digital generators that can be used in generation of digital fuzzy parametric conjunctions is proposed. These generators together with basic t-norms and t-conorms can be used to generate new digital fuzzy conjunctions with effective hardware implementation. A new method of generation of digital fuzzy parametric conjunctions is considered. An example of FPGA implementation of new fuzzy parametric conjunctions is done.

I. INTRODUCTION

Hardware implementation of fuzzy systems plays an important role in industrial applications of fuzzy logic based intelligent systems [1-4]. Digital fuzzy hardware systems became more popular than analogical fuzzy systems. Digital systems have several advantages [5]: digital systems are generally easier to design, it is easy to store digital information, accuracy and precision is greater etc. Parameterization of fuzzy systems facilitates adjustment of fuzzy systems by tuning parameters of the system. Usually it is used a parameterization of membership functions of linguistic variables [14] but an optimization of fuzzy systems by parameters of fuzzy sets can cause considerable distortion of initially given membership functions and as a result a meaning of fuzzy sets can be changed and expert knowledge given in fuzzy variables can be lost. Alternatively it can be used a parameterization of fuzzy operations and tuning of parameters of these operations can be done [6-8]. 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 digital fuzzy parametric operations that can be efficiently implemented in hardware. In [9] the methods of generation of parametric families of fuzzy conjunctions based on generators that have simple hardware implementation were proposed. In this paper we propose new methods of generation of parametric families of fuzzy conjunctions simple for hardware implementation. This approach uses simple parametric generators together with min-max, Lukasiewicz and drastic t-norms and corresponding t-conorms [10] that have efficient digital hardware implementation. These t-norms, t-conorms and generators can be used as bricks for construction new digital fuzzy parametric operations. Several simple parametric classes of digital fuzzy conjunctions generated in such manner are considered. An example of circuits for hardware implementation of new digital fuzzy parametric conjunction is given.

The paper has the following structure. Section 2 gives the basic definitions of fuzzy conjunction and disjunction operations. Section 3 presents a wide class of simple digital generators that can be used together with basic t-norms and t-conorms in generation of fuzzy parametric conjunctions. In Section 4 a new method of generation of fuzzy parametric conjunctions suitable for hardware implementation is proposed. Examples of a FPGA implementation of proposed parametric conjunction operations are considered is Section 5. In Conclusion we discuss obtained results and future directions of research.

II. BASIC DEFINITIONS

In binary logic, where the set of true values contains only two elements $L = \{0, 1\}$, the negation $\neg$, conjunction $\land$ and disjunction $\lor$ operations are defined as follows: $\neg 0 = 1$, $\neg 1 = 0$, $0 \land 0 = 0$, $1 \land 0 = 0$, $0 \land 1 = 1$, $0 \lor 0 = 0$, $1 \lor 0 = 1$, $0 \lor 1 = 1$, $1 \lor 1 = 1$. 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 $N:L \rightarrow L$, $T:L \times L \rightarrow L$ and $S:L \times L \rightarrow L$. Denote the maximal true value as follows: $I = 1$, then the conditions considered above have the form:

\[
N(0) = 1, \quad N(1) = 0. \quad (1)
\]

\[
T(0,0) = 0, \quad T(1,0) = 0, \quad T(0,1) = 0, \quad T(1,1) = 1. \quad (2)
\]

\[
S(0,0) = 0, \quad S(1,0) = 1, \quad S(0,1) = 1, \quad S(1,1) = 1. \quad (3)
\]

In most general form fuzzy negation, conjunction and disjunction operations can be defined as functions satisfying these conditions [11] but such definitions do not say anything about properties of these operations on the set of values $(0, 1)$. For this reason, the most common definitions of these operations include monotonicity properties:

\[
N(x) \geq N(y), \quad \text{if} \quad x \leq y, \quad (4)
\]

\[
T(x,y) \leq T(u,v), \quad S(x,y) \leq S(u,v), \quad \text{if} \quad x \leq u, \quad y \leq v. \quad \text{(monotonicity)} \quad (5)
\]
Fuzzy negation is defined as a function $N:L \rightarrow L$, satisfying (1) and (4). A fuzzy negation is called an involution if it satisfies the following condition:

$$N(N(x)) = x.$$  

(6)

Axioms (2),(3),(5) define an axiomatic skeleton [12] for fuzzy conjunction and disjunction operations. In [8], such generalized operations are called $G$-conjunction and $G$-disjunction operations. It can be shown that these operations satisfy the following conditions:

$$T(x,0) = 0, \quad T(0,y) = 0,$$

$$S(x,1) = 1, \quad S(1,y) = 1.$$  

(7)

(8)

In [7], functions $T$ and $S$ satisfying (5),(7),(8) are called pseudo-conjunctions and pseudo-disjunctions, respectively. In [6,7], fuzzy conjunction and disjunction operations are defined by monotonicity (5) and boundary conditions:

$$T(x,1) = x, \quad T(1,y) = y, \quad (\text{boundary conditions})$$  

$$S(x,0) = x, \quad S(0,y) = y. \quad (\text{boundary conditions})$$  

(9)

(10)

This definition coincides with the definition of t-seminorms and t-semicnors in [13]. It is clear that fuzzy conjunctions and disjunctions satisfy (2), (3) and (7), (8). t-norms and t-conorms [10] are defined by (9), (10), (5), commutativity and associativity axioms. Below are the most simple pairs of t-norms and t-conorms:

$$T_{df}(x,y) = \min\{x,y\}, \quad (\text{minimum}),$$  

(11a)

$$S_{df}(x,y) = \max\{x,y\}, \quad (\text{maximum}),$$  

(11b)

$$T_{1}(x,y) = \max\{x+y-1,0\}, \quad (\text{Lukasiewicz}),$$  

(12)

$$S_{1}(x,y) = \min\{x+y,1\}, \quad (\text{Lukasiewicz}),$$  

(12a)

$$T_{d}(x,y) = \begin{cases} x, & \text{if } y = I \\ y, & \text{if } x = I \\ 0, & \text{if } x, y < I \end{cases}, \quad (\text{drastic product}),$$  

(13)

$$S_{d}(x,y) = \begin{cases} x, & \text{if } y = 0 \\ y, & \text{if } x = 0 \\ I, & \text{otherwise} \end{cases}, \quad (\text{drastic sum}),$$  

(14)

Lukasiewicz t-norm and t-conorm are also known as a bounded product and a bounded sum, respectively [14].

Considered above three pairs of t-norms and t-conorms are composed from mathematical operations min, max, bounded sum, bounded difference and comparison that have efficient digital hardware implementation [15]. These t-norms and t-conorms will be called basic fuzzy conjunction and disjunction operations. In the following sections these operations will be used for generation of parametric classes of conjunction and disjunction operations also simple for hardware implementation. Note that the product t-norm $T_{d}(x,y) = xy$, and the probabilistic sum $S_{p}(x,y) = x+y-xy$ have less efficient digital hardware implementation than basic operations (14)-(17) due to the presence of product operation [16-17].

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

$$S(x,y) = N(T(N(x),N(y))), \quad (15a)$$

$$T(x,y) = N(S(N(x),N(y))). \quad (15b)$$

For this reason we will consider mainly fuzzy conjunctions.

It can be shown [7] that fuzzy conjunctions and disjunctions satisfy the following inequalities:

$$T_{d}(x,y) \leq T(x,y) \leq T_{d}(x,y) \leq S_{d}(x,y) \leq S_{d}(x,y) \quad (16)$$

As in [6-8], fuzzy conjunction and disjunction operations will be defined in this paper as functions $T$ and $S$ satisfying axioms (5), (9), (10). Commutative and associative fuzzy conjunction and disjunction operations (t-norms and t-conorms) are very important in mathematical applications of fuzzy logic but in a wide class of fuzzy systems these properties are not used (see [6-8] for motivations). Moreover, known parametric classes of t-norms and t-conorms are usually complicated for tuning in fuzzy systems and for hardware implementation.

Several methods of generation of simple parametric fuzzy operations suitable for tuning in fuzzy systems have been proposed in [6-8]. These methods are based on the formula:

$$T(x,y) = T_{2}(T_{1}(x,y),s(x,y)),$$  

(17)

where $T_{2}$ and $T_{1}$ are conjunctions and $s$ is a pseudo-disjunction. Suppose $h,g_{1},g_{2}:L \rightarrow L$ are non-decreasing functions called generators and $g_{1}$ and $g_{2}$ satisfy conditions: $g_{1}(1) = g_{2}(1) = 1$. Pseudo-disjunction $s$ can be generated by means of generators and other pseudo-disjunctions $s_{2}$ and $s_{1}$ as follows:

$$s(x,y) = s_{1}(g_{1}(x),g_{2}(y)),$$  

(18)

$$s(x,y) = g_{1}(s_{1}(x,y)),$$  

(19)

$$s(x,y) = s_{2}(s_{1}(x,y),h(y)).$$  

(20)
As pseudo-disjunctions s in (17) one can use basic disjunctions or pseudo-disjunctions obtained from basic disjunctions by recursive application of (18)-(20). Below are examples of simplest parametric conjunctions obtained in [7] by means of (17)-(20):

\[ T(x,y) = \min(x,y) \max(l-p(1-x),I-q(1-y),0). \quad (21) \]

\[ T(x,y) = \min\{\min(x,y), \max(xp,yp)\}. \quad (22) \]

These operations are simple for tuning in fuzzy models but they have not efficient hardware implementation because they are based on product or exponentiation functions used in generators.

The goal of this paper is to introduce a wide class of digital fuzzy parametric conjunctions which have efficient hardware implementation and can be assembled in integrated circuit containing some basic components such that a suitable selection of a combination of these components will define some fuzzy parametric conjunction. As a general scheme for generating different fuzzy parametric conjunctions depending on suitable selection of components it can be used the set of formulas (17)-(20) containing as components t-norms, t-conorms and generators. These components can serve as bricks for construction digital fuzzy parametric conjunctions. Basic t-norms and t-conorms considered above have efficient hardware implementation. In the following section we introduce a set of digital generators having efficient hardware implementation that can be used in (17)-(20). These generators in their definition use only functions that have efficient hardware implementation such as constant, identity, comparison, maximum and minimum, bounded sum and bounded difference. Further we introduce a new method of generation of digital fuzzy parametric conjunctions containing as components basic t-norms, t-conorms and generators.

III. DIGITAL REPRESENTATION OF MEMBERSHIP VALUES, GENERATORS AND FUZZY OPERATIONS

Suppose it is used \( m \) bits in digital representation of membership values. Then different membership values can be presented by \( 2^m \) numbers from the set \( L = \{0,1,2,\ldots,2^m-1\} \). Denote the maximal membership value \( I = 2^m-1 \). This value will represent the full membership in a traditional set of membership values \( L = [0,1] \). All definitions and properties of fuzzy operations (1)-(20) from the previous section can be transformed into the digital case by replacing the set of membership values \( L = [0,1] \) by \( L = \{0,1,2,\ldots,2^m-1\} \) and maximal membership value \( I = 2^m-1 \). For graphical representation of digital generators and fuzzy operations we will use below \( m = 4 \) bits, with the following set of digital membership values \( L = \{0,1,2,\ldots,14,15\} \) and with the maximal membership value \( I = 15 \). Fig.1 depicts basic t-norms and t-conorms in digital form.

To generate by means of (17)-(20) new parametric conjunctions that have efficient digital hardware implementation we need to introduce generators that use in their definition only basic functions which have efficient digital hardware implementation: constant, identity, comparison, minimum, maximum, bounded sum and bounded difference. It is clear that the basic t-norms and t-norms use only these functions in their definition. Below we introduce generators composed from basic functions and containing some parameter \( p \). We consider here only simplest generators depending only on one parameter \( p \). We suppose that parameter \( p \) can change from 0 till 1. In some cases we use the constants corresponding to the value 0.5 in true scale [0,1]. Because the set of digital true values \( L = \{0,1,2,\ldots,2^m-1\} \) contains an even number of values then as a constant corresponding to 0.5 in traditional scale we can use 2 constants denoted as \( L_{0.5} \) (left 0.5) and \( R_{0.5} \) (right 0.5) and defined as follows: \( L_{0.5} = 2^{m \cdot 1} \), \( R_{0.5} = 2^{m \cdot 1} \). For example, for \( m = 4 \) bits we have \( I = 2^m - 1 = 15 \), \( L_{0.5} = 7 \) and \( R_{0.5} = 8 \). Some simple digital generators were introduced in [9]. Below we introduce an extended list of simple generators that have efficient hardware implementation. To make references on these generators we use suitable names of them that pointed out in the right side of the definition of generator. Note that these names slightly differ from the names of corresponding generators considered in [9].

\[ g(x) = x, \quad <x> \]

\[ g(x) = p, \quad <p> \]

\[ g(x) = \begin{cases} 0, & \text{if } x \leq p \\ I, & \text{otherwise} \end{cases}, \quad <0,I> \]

\[ g(x) = \begin{cases} p, & \text{if } x \leq p \\ I, & \text{otherwise} \end{cases}, \quad <p,I> \]

\[ g(x) = \begin{cases} 0, & \text{if } x \leq p \\ p, & \text{otherwise} \end{cases}, \quad <0,p> \]

\[ g(x) = \begin{cases} 0, & \text{if } x \leq p \\ p, & \text{if } p < x \leq I - p \\ I, & \text{otherwise} \end{cases}, \quad <0,p,I> \]

\[ g(x) = \begin{cases} 0, & \text{if } x \leq p \\ x, & \text{otherwise} \end{cases}, \quad <0,x> \]

\[ g(x) = \begin{cases} x, & \text{if } x \leq p \\ I, & \text{otherwise} \end{cases}, \quad <x,I> \]

\[ g(x) = \begin{cases} p, & \text{if } x \leq p \\ x, & \text{otherwise} \end{cases}, \quad <p,x> \]
conjunction obtained by (17) and (18) by means of basic t-interval \([0,1]\).\
Fig. 3 shows an example of digital fuzzy parametric conjunction obtained by (17) and (18) by means of generator \(<x,p>\) and basic fuzzy conjunction and disjunction operations: \(T_M^1 = T_M, T_M^2 = T_L, s = S_M\). The value of the parameter \(p\) in generator is equal to 3 in digital scale.

Fig. 4 contains an example of digital fuzzy parametric conjunction obtained by (17) and (19) by means of generator \(<x,p,l>\) and basic fuzzy conjunction and disjunction operations: \(T_M^1 = T_M, T_M^2 = T_L, s = S_M, s_2 = S_D\).

IV. NEW METHOD OF GENERATION OF FUZZY PARAMETRIC CONJUNCTIONS

Here we propose a new method of generation of fuzzy parametric conjunctions. This method together with the method considered in previous sections can be used to generate the library of digital fuzzy parametric conjunctions that can be implemented in integrated circuits in digital hardware implementation of fuzzy systems.

Theorem 1. Suppose \(T_1\) and \(T_2\) are fuzzy conjunctions, \(S\) is a fuzzy disjunction, and \(g\) is a generator then the formula

\[T(x,y) = \min(T_1(x,y), S(T_2(x,y), g(y))), \quad (23)\]

defines a fuzzy conjunction \(T\).

We will write \(T_1 \leq T_2\) if \(T_1(x,y) \leq T_2(x,y)\) for all \(x, y \in [0,1]\).

Proposition 2. For specific conjunctions \(T_1\) and \(T_2\) a conjunction \(T\) in (23) is reduced as follows: if \(T_1 \leq T_2\) then \(T = T_1\).

From Proposition 2 and ordering of t-norms and t-conorms (16) it follows that to obtain \(T\) we can to use only the following pairs of conjunctions: \((T_1 = T_M, T_2 = T_D)\) and \((T_1 = T_M, T_2 = T_L)\). Fig. 6 contains an example of parametric fuzzy conjunction obtained by (23) by means of generator \(<0,p>\) and basic fuzzy conjunctions and disjunctions: \(T_M^1 = T_M, T_M^2 = T_L, s = S_M, s_2 = S_D\).

V. HARDWARE IMPLEMENTATION OF DIGITAL FUZZY PARAMETRIC CONJUNCTIONS

A synthesis of circuits was realized by a Spartan 3E 3S500EFG320-5 FPGA from Xilinx using ISE Webpack tools for gate level design and digital circuitry knowledge [5]. Basic blocks used have eight bit data buses, and they are a part of system libraries with exception of multiplexer which was constructed using logical gates. The following common constructs are used [16,17]:

Two eight bit data full adder with eight bit data result and carry output (ADD8).
Two eight bit data subtractor with eight bit data result and overflow output (ADSUB).

Two eight bit data comparator with one bit outputs LESS THAN and GREATER THAN (COMPMC8).

Two eight bit data inputs and one eight bit data result with one bit selector (MUX2_8B).

In order to realize an efficient digital implementation of fuzzy parametric conjunctions it is necessary to use simple operations as mentioned in Sections 2 and 3. Consider for x and y inputs of these operations an integer 8 bit universe. As a simple example consider the circuits that can be used to implement fuzzy parametric conjunction presented in Fig. 6:

\[ T(x,y) = T_M(T_M(x,y),S_M(T_L(x,y),g(y))), \]

where a generator \(<0,p>\) is used.

The first block required is a minimum \(T_M\) conjunction operation with two eight bit inputs as shown in Fig. 7. Resultant circuit is realized using an eight bit numbers comparator and multiplexer. The output GT of comparator is used as input selection SEL for a multiplexer. When \(A\) is greater than \(B\) the output GT obtains logic value ONE that gives to the minimum value \(B\) to pass as a result of the circuit. In opposite case when \(A\) is less than \(B\), GT send logical value ZERO that gives to the minimum value \(A\) to pass as a result.

A circuit for Lukasiewicz conjunction \(TL\) is presented in Fig. 8. It contains eight bit adder, subtractor and multiplexer. Adder and subtractor are used to calculate \(x+y-I\). Carry output from adder is used as input selection SEL for a multiplexer. When sum of \(x\) and \(y\) generates a carry it means that the result is big enough to have a positive result when \(I\) is subtracted and it is a valid result. In the case when sum of \(x\) and \(y\) does not generate a carry, the result of a subtraction will be negative and ZERO must be lead to the output of multiplexer.

Fig. 9 depicts a maximum SM disjunction operation between two eight bit numbers. A circuit is similar to the circuit for minimum operation presented in Fig. 7.

The circuit for \(<0,p>\) generator similarly to the circuits of minimum and maximum operations contains one comparator and one multiplexer such that the inputs of multiplexer are SEL, \(p\) and 0.

Fig. 10 depicts a circuit implementing fuzzy parametric conjunction (24). This circuit composed of circuits of t-norms, t-conorm and generator used in construction of this conjunction. These circuits are denoted as blocks MINIMO for TM, MAXIMO for SM, BOND_PROD for TL, and \(<0,p>\) for corresponding generator.

Fig.11 presents simulation results for integrated circuit shown in Fig.10 for parameter value \(p=102\). Tab.1 presents resources consumed by obtained hardware.

VI. CONCLUSIONS

The paper proposes the methods of generation of a wide class of digital fuzzy parametric conjunctions which have efficient hardware implementation. First, the traditional set on membership values \(L=\{0,1\}\) is replaced by the digital set of true values \(L=\{0,1,2,\ldots,n\}\), where \(I=2^n-1\) and \(n\) is a number of bits used in presentation of membership values. Such replacement of the set of true values on the one hand preserves most of definitions and properties of fuzzy operations that can be used in digital representation of membership values and fuzzy logic operations. On the other hand digital representation of true values simplifies digital hardware implementation of fuzzy logic operations and gives more convenient form for reasoning and manipulation with digital generators and parameters.

Second, from the point of view of hardware implementation of digital systems it is important to have the basic blocks than can be used as bricks in construction of all system. The methods of generation of fuzzy parametric conjunctions considered here are based on the same idea to use simple basic functional blocks to construct more complicated fuzzy parametric conjunctions. As such blocks they are used basic t-norms and t-conorms together with the set of simple generators. These basic functions have simple and effective implementation as digital hardware blocks that can be easy integrated to realize fuzzy parametric conjunctions.

The obtained results can be extended in several directions. Digital fuzzy parametric disjunctions can be constructed from fuzzy conjunctions by means of suitable negation operation. 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 system in control, pattern recognition and decision making.

ACKNOWLEDGMENT

The research work was partially supported by IMP project D.00507, Bilateral CONACYT-NKTH project No. II101/127/08 Mod. Ord. 38/08, by ICyTDF funding, award No. PICCT08-22, and by matching funding by IPN, award No. SIP/DF/2007/143.

REFERENCES

Figure 1. Basic t-norms and t-conorms in digital representation with 4 bits
Figure 2. Generators with parameter value $p = 3$ in digital representation with 4 bits.
Figure 3. Example of digital fuzzy parametric conjunction obtained by (17) and (18)

Figure 4. Example of digital fuzzy parametric conjunction obtained by (17) and (19)

Figure 5. Example of parametric fuzzy conjunction obtained by (17) and (20)

Figure 6. Example of fuzzy parametric conjunction obtained by (23)
Figure 7. A circuit for the minimum conjunction operation

Figure 8. A circuit for Lukasiewicz conjunction operation

Figure 9. A circuit for maximum disjunction operation

Figure 10. A circuit for implementation of fuzzy parametric conjunction (24)
Figure 11. Simulation results for circuit presented in Fig. 10

<table>
<thead>
<tr>
<th>TABLE I. HARDWARE RESOURCES CONSUMED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic gates</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>382</td>
</tr>
</tbody>
</table>