Int. J. Fuzzy Syst. (2015) 17(4):559–576
DOI 10.1007/s40815-015-0076-6

Hardware Design of Digital Parametric Conjunctors and t-Norms

Prometeo Cortés-Antonio[1] · Ildar Batyrshin[1] · Luis A. Villa-Vargas[1] · Imre Rudas[2] · Herón Molina-Lozano[1] · Marco A. Ramírez-Salinas[1]

Received: 3 April 2015 / Revised: 26 June 2015 / Accepted: 14 August 2015 / Published online: 9 September 2015

Abstract This paper presents the hardware design and its implementation on FPGA of several parametric families of digital conjunctors and t-norms built from simple basic t-norms. The authors propose the method of unified presentation of the p-monotone sum, the simplified versions of the ordinal sum of t-norms and t-subnorms, and the method of extension of t-norms by the drastic t-norm. Such unification gives possibility to join several methods of construction of parametric digital conjunctors and t-norms in one scheme with the efficient FPGA implementation. The logic schemes of the proposed design are presented, and the comparative analysis of the latency time and the resources used for the implementation is given.

Keywords Fuzzy logic · Conjunctor · t-Norm · Monotone sum · Ordinal sum · FPGA

✉ Ildar Batyrshin
batyr1@gmail.com

Prometeo Cortés-Antonio
prometeo.cortes@gmail.com

Luis A. Villa-Vargas
lvilla@cic.ipn.mx

Marco A. Ramírez-Salinas
mars@cic.ipn.mx

[1] Centro de Investigación en Computación, Instituto Politécnico Nacional, México, DF, México

[2] Obudu University, Budapest, Hungary

1 Introduction

Fuzzy logic [1] has wide applications in industry and in decision-making systems [2, 3]. Many of them are based on hardware implementation of fuzzy systems [4–7]. Recent years, the number of applications of fuzzy systems developed on field programmable gate array (FPGA) has been increased [7–13]. FPGAs are programmable semiconductor devices that are based around a matrix of configurable logic blocks connected through programmable interconnects [14]. FPGAs can be easily reprogrammed to the desired application or functionality requirements [10, 14–20]. This paper presents the design in one scheme of FPGA of several parametric families of conjunctors and t-norms built from simple basic t-norms that can be used in efficient FPGA implementation of reconfigurable fuzzy systems. Below is a short survey of the works on parameterized and generalized fuzzy operations and their FPGA implementation.

The generalized operations of fuzzy logic-like non-associative or non-commutative conjunctions (conjunctors) and parameterized or reconfigurable t-norms give possibility to build sophisticated and more flexible fuzzy models in comparison with traditional fuzzy models based on basic t-norms [21–24]. The most general extension of conjunction operation T from binary to fuzzy setting requires the monotonicity of this operation and the fulfillment of boundary conditions [25]: $T(0,0) = T(1,0) = T(0,1) = 0$, $T(1,1) = 1$. The most important from the mathematical point of view fuzzy extension of conjunction operation is a t-norm [26, 27] satisfying also the associativity and commutativity properties and the boundary condition $T(1,x) = x$ for all x in [0, 1]. But the hardware implementation of the most popular parametric families of (associative) t-norms consumes sufficiently many resources due to their use in their definition of the multiplication,

division, exponentiation operations, or logarithm function. As it was noted in [23, 24], the associativity property often does not require from the fuzzy conjunction operations used in applied fuzzy models with two inputs. It gives rise to build more simple non-associative parametric conjunction operations suitable for efficient hardware implementation. The commutativity property of conjunctions also may be not necessary when the position of arguments in fuzzy models is fixed. For these reasons, the study of non-associative and non-commutative conjunctions (conjunctors) has the practical interest in fuzzy modeling. Another point of interest in building of fuzzy models is to consider parameterized fuzzy operations that can be tuned to achieve the better performance of these models [23, 24]. The tuning of membership functions in fuzzy models can lead to fuzzy sets that loss the expert knowledge presented by fuzzy sets before the tuning. In such cases, the tuning of parameters of fuzzy operations instead of (or additionally to) tuning of membership functions can help to preserve expert knowledge presented in fuzzy sets. The main parametric families of t-norms and t-conorms are described in [26]. In [23, 24], the simple parametric classes of fuzzy conjunctors suitable for tuning in fuzzy models are introduced. In [28], it compared the influence of fuzzy control systems on the tuning parameters of operations versus the parameters of membership functions. The paper [29] discussed the methods of information aggregation in intelligent systems using generalized operators. The application of generalized operations of fuzzy logic in inference engines is discussed in [30]. Zhang et al. [31] studied a fuzzy logic system based on parametric family of Schweizer–Sklar t-norm. Different aggregation functions including generalized conjunctions are considered in [32]. Alcalá-Fdez [33] used parameterized operations for achieving better co-operation among fuzzy rules. The methods of construction of parametric classes of digital conjunctors based on generator functions using simple operations like addition, subtraction, minimum, maximum, and comparison suitable for efficient hardware implementation are introduced in [34–36]. The digital conjunctors considered in these works are particular cases of discrete fuzzy conjunctors [37, 38] defined on a set of integer numbers $L = \{0, 1,\ldots, n\}$. In [39], it proposed the method of the monotone sum of basic t-norms for the construction of simple parametric conjunctors for hardware implementation. The paper [40] presents modular neuro-fuzzy systems based on parametric classes of generalized conjunction and disjunction operations. A survey of weak connectives and the problem of the preservation of their properties in some aggregation functions are considered in [41]. Dependencies between fuzzy conjunctions and implications are studied in [42, 43]. Application of parametric t-norms and various fuzzy operations in neural networks is considered in [44, 45]. t-norms and t-conorms on the sets of multisets or strings to achieve strict monotonicity of these operations for finite scales are considered in [46]. In [47], the Hamacher parameterized t-norms are used to induce the priority weight in prioritized weighted aggregation in multicriteria decision making. The authors of [48] used parametric conjunctors in a fuzzy rule-based control system. The paper [49] used parameterized uninorm and absorbing norm for logic design. The tuning of parameterized conjunctors is used in the optimization of the type-l fuzzy neural system for slip control of a quarter car model [50], for trajectory tracking of a 2-DOF helicopter system using neuro-fuzzy system [51], and for optimization of an interval type-2 fuzzy neural system [52]. The paper [53] studied a general class of increasing binary operations including conjunctions. The application of generalized norms in digital/analog scheme's synthesis and analysis is discussed in [54]. The parameterized operations over t-norms are considered in [55]. Adaptive conjunction operations are used in [56] in linguistic fuzzy modeling. The paper [57] discussed the relationship between monotone and ordinal sums of basic t-norms and proposed the new methodologies for generation of parametric digital t-norms suitable for efficient hardware implementation. The parametric inverse Hamacher operations are discussed in [58]. The authors of [59] proposed a generalized fuzzy similarity measure that can be used in the construction of fuzzy systems. The paper [60] developed multivalued automaton based on generalized fuzzy operations for control of mobile agents and the locomotion model. The application of the generalized logic operations in fuzzy predicate systems for clustering is discussed in [61]. Preservation of fuzzy relation properties based on generalized fuzzy conjunctions and disjunctions during aggregation process is studied in [62]. FPGA implementation of different parametric families of digital conjunctors based on generator functions is considered in [63–65]. FPGA implementation of diverse fuzzy t-norms and t-conorms is considered in [66]. The papers [67, 68] proposed the methods of FPGA implementation of p-monotone sum and $(p, 1 - p)$ monotone sum of basic t-norms. FPGA implementation of Sugeno and Mamdani fuzzy inference systems with parametric families of conjunctors obtained by monotone sum of basic t-norms is discussed in [69, 70].

This work is based on the theoretical results from [57]. The main contribution of the paper is the following. The paper proposes the method of unified presentation of the p-monotone sum, the simplified versions of the ordinal sum of t-norms and t-subnorms, and the method of extension of t-norms by the drastic t-norm. Such unification gives possibility to join several methods of construction of parametric digital conjunctors and t-norms in one scheme with the efficient FPGA implementation. The logic schemes of

the proposed design are presented, and the comparative analysis of the latency time and the resources used for the implementation is given.

The paper has the following structure. Section 2 gives the definitions of digital conjunctors and t-norms. Section 3 discusses the methods of construction of parametric conjunctors and t-norms implemented further on FPGA. Section 4 presents the design on FPGA of parametric families of conjunctors and t-norms. Section 5 contains the comparative analysis of the latency time and the resources used for the implementation. The last section contains conclusions.

2 Digital Conjunctors and t-Norms

Consider the set $L = \{0, 1,\ldots, n\}$, $n \geq 1$ of digital representations of truth (degree, membership, etc.) values. If m-bits integer representation is used, then we have $n \leq 2^m - 1$. We will consider conjunctors and t-norms defined on the set L, and the maximal value of this set $\mathbf{I} = n$ will correspond to the value 1 in traditional set of truth values [0, 1]. Consider the function $T{:}L \times L \to L$ with the following possible properties on L:

A1. $T(x, \mathbf{I}) = x$, $T(\mathbf{I}, y) = y$, (boundary conditions)

A2. $T(x, y) \leq T(u, v)$, if $x \leq u$, $y \leq v$. (monotonicity)

A3. $T(x, y) = T(y, x)$, (commutativity)

A4. $T(x, T(y, z)) = T(T(x, y), z)$. (associativity)

A5. $T(x, y) \leq \min(x, y)$. (range condition)

This function will be called a *conjunctor*, a *commutative conjunctor*, a *t-norm*, or a *t-subnorm,* if the following properties are fulfilled for all $x, y, z \in L$, correspondingly {A1, A2}, {A1–A3}, {A1–A4}, and {A2–A5}. See [26, 27, 37, 38, 57] for properties of these operations on [0, 1], on discrete set and on $L = \{0, 1,\ldots, n\}$. For any conjunctor from A1, A2, it follows for all $x, y \in L$:

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

Generally, we can consider conjunctors and t-norms defined on a set L containing only one element $L = \{0\}$. In this case, we have $\mathbf{I} = 0$. Below there are the simplest t-norms that will be considered as basic t-norms in the generation of digital fuzzy parametric conjunctors and t-norms:

$$T_M(x, y) = \min\{x, y\}, \quad \text{(Minimum)}$$

$$T_N(x, y) = \begin{cases} \min(x, y), & \text{if } x + y > \mathbf{I} \\ 0, & \text{otherwise} \end{cases}$$

(Nilpotent minimum)

$$T_L(x, y) = \max\{x + y - \mathbf{I}, 0\},$$

(Lukasiewicz t−norm)

$$T_D(x, y) = \begin{cases} x, & \text{if } y = \mathbf{I} \\ y, & \text{if } x = \mathbf{I} \\ 0, & \text{if } x, y < \mathbf{I} \end{cases}. \quad \text{(Drastic product)}$$

These t-norms have efficient hardware implementation, because their definition uses very simple mathematical operations. Figure 1 depicts the shapes of these t-norms defined on $L = \{0, 1, ..., 31\}$..

In some cases, we will also consider a basic conjunctor as the following operation:

$$T_P(x, y) = x * y, \quad \text{(product)}$$

where the operation * is a normalized product of x and y and taking values in L:

$$T_P(x, y) = xy/\mathbf{I}.$$

This operation has a little bit more complicated hardware implementation on the set of integer values L than other basic t-norms; moreover, this digitalized version of the product t-norm will be not associative and hence will be not a t-norm but only a conjunctor. It can be used as a digital approximation of the product t-norm.

Denote $T_1 \leq T_2$ if $T_1(x,y) \leq T_2(x,y)$ for all x,y from L. For any conjunctor T and for considered above t-norms, we have

$$T_D \leq T \leq T_M, \quad T_D \leq T_L \leq T_N \leq T_M, \quad T_D \leq T_L \leq T_P \leq T_M.$$

From $T \leq T_M$, it follows that any t-norm is a t-subnorm.

3 Construction of Parametric Conjunctors and t-Norms

Let $L = \{0, 1,\ldots, n\}$ be a set of integer values and $a, b \in L$, be two integers such that $a \leq b$. A sequence of consecutive numbers changing from a till b will be called an interval and denoted as $[a, b]$. Let $p \in \{0, 1,\ldots, n\}$ is an integer parameter. When $p \geq 1$ divide L on 2 intervals: $X_1 = [0, p - 1]$, $X_2 = [p, n]$, and all domain $L \times L$ divide on 4 sectors: $D_1 = X_1 \times X_1$, $D_2 = X_2 \times X_1$, $D_3 = X_1 \times X_2$, $D_4 = X_2 \times X_2$, see Fig. 2 for $n = 15$. When $p = 0$ we have only one interval $X_2 = [0, n]$ and one sector $D_4 = X_2 \times X_2 = L \times L$ coinciding with all domain $L \times L$. In this case, all other sectors D_1, D_2, and D_3 will be empty.

Below we consider methods of construction and hardware implementation of conjunctors and t-norms on $L \times L$ by means of basic t-norms T_i associated with sectors D_i, $i \in \{1, 2, 3, 4\}$. These methods are particular cases of the general methods discussed in [26, 27, 37, 38, 57]. To simplify references on these methods, we give short names for them.

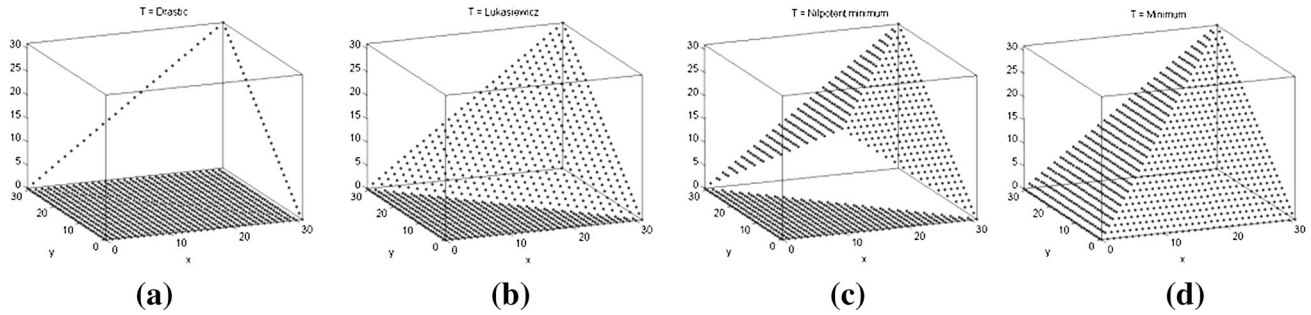

Fig. 1 Simplest t-norms: **a** Drastic, **b** Lukasiewicz t-norm, **c** Nilpotent minimum, **d** Minimum

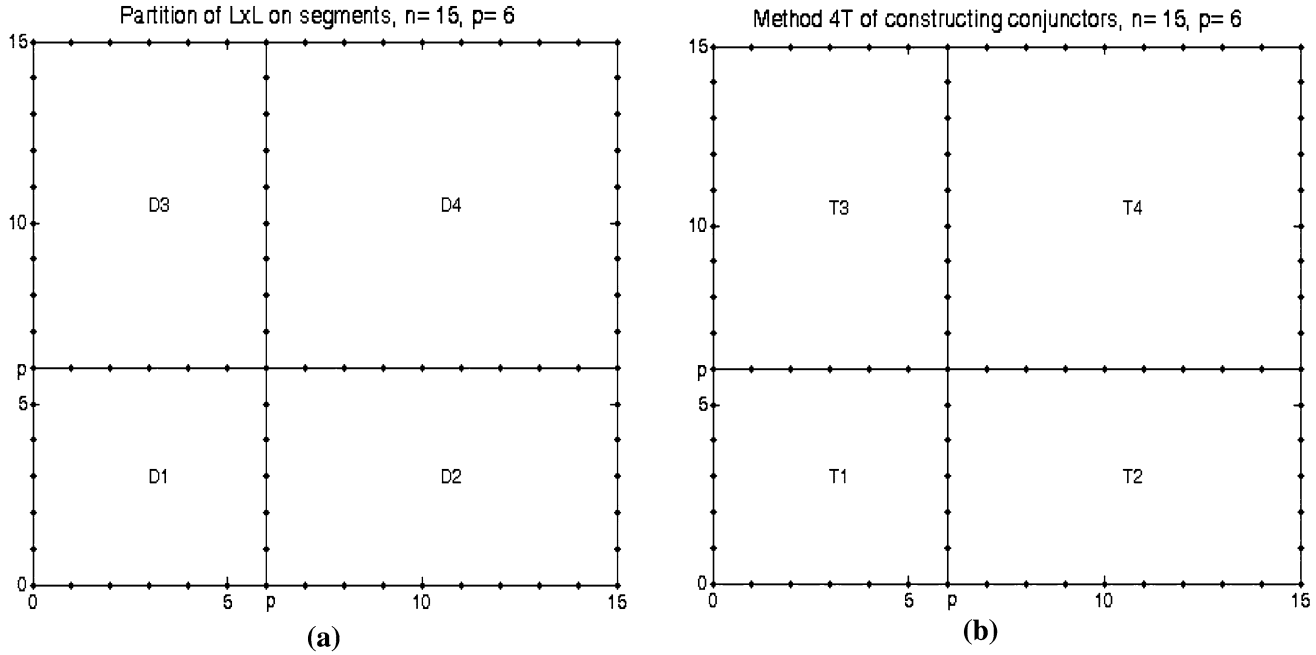

Fig. 2 **a** Partition of $L \times L$ on segments defined by parameter p. **b** Method 4T of constructing conjunctors

3.1 Method 4T

This method can use four different t-norms. Let T_i, $i \in \{1, 2, 3, 4\}$ be t-norms defined on $L = \{0,1,\ldots, n\}$, such that

$$T_1 \le T_2 \le T_4, \quad T_1 \le T_3 \le T_4.$$

Then, the function T: $L \times L \to L$ defined by

$$T(x,y) = \begin{cases} T_1(x,y), & \text{if } x,y<p \\ T_2(x,y), & \text{if } y<p, p \le x, \\ T_3(x,y), & \text{if } x<p, p \le y \\ T_4(x,y), & \text{if } p \le x,y \end{cases},$$

is a conjunctor, where $p \in \{0, 1,\ldots, n\}$ is a parameter. Figure 2b shows t-norms used in segments $D_1 - D_4$ of $L \times L$ in construction of conjunctor $T(x,y)$ by method 4T. Conjunctor $T(x,y)$ is commutative if all T_i are commutative and $T_2 = T_3$. Note that when $p = 0$, the conjunctor T equals to T_4 defined on all domain $L \times L$. On the other hand, when $p = n$, the resulting conjunctor T equals to T_1 because T used only border values of t-norms T_2, T_3, and T_4 defined by **A1** condition and hence coinciding with the corresponding values of T_1.

The method 4T of generation of conjunctors is based on the p-monotone sum of t-norms [39, 57]. We denote here this method for short as 4T instead of (T_1, T_2, T_3, T_4) used in [57] and we use notation 4Tc for commutative conjunctors. In Table 1 and Fig. 6, this method has types 0 and 1 for non-commutative conjunctors and types 2 and 3 for commutative conjunctors. Similarly, below, instead of the notations (*T, M, M, M*), (*D, D, D, T*), (*T, M, M, T*), used in [57] where M denotes T_M and D denotes T_D we use notations T3M, 3DT, and TMMT, respectively.

Table 1 Classification of conjunctors and t-norms by type configuration parameter

Type code	Type methodology
0, 1	Non-commutative 4T
2, 3	Commutative 4Tc
4	3DT
5	3DTs
6	TMMT
7	TMMTs

3.2 Method 3DT

Suppose T_D is defined on $L = \{0, 1,\ldots, n\}$, $p \in \{0, 1,\ldots, n\}$ and t-norm T_4 is defined on $L_4 = \{0, 1,\ldots, n - p\}$, then for all $x, y \in L = \{0, 1,\ldots, n\}$ the following function is a t-norm on L:

$$T(x,y) = \begin{cases} p + T_4(x-p, y-p), & \text{if } p \le x, y \\ T_D(x,y), & \text{otherwise} \end{cases}.$$

The method 3DT is based on Proposition 3 from [57]. Figure 3a depicts the location of basic t-norms used in this method in segments of $L \times L$. In Table 1 and Fig. 6, this method is referred to as the method of type 4. See Table 10 and Fig. 10 for examples.

3.3 Method TMMT

Let $p \in \{0, 1,\ldots, n\}$, T_1 be a t-norm on $L_1 = \{0, 1,\ldots, p\}$ and T_4 be a t-norm on $L_4 = \{0, 1,\ldots, n - p\}$, then the following function is a t-norm on $L = \{0, 1,\ldots, n\}$:

$$T(x,y) = \begin{cases} T_1(x,y) & \text{if } x, y < p \\ p + T_4(x-p, y-p) & \text{if } p \le x, y \\ \min(x,y) & \text{otherwise} \end{cases}.$$

This method is based on ordinal sum of t-norms [26, 37, 38, 57]. Figure 3b depicts the location of basic t-norms used in this method in segments of $L \times L$. In Table 1 and Fig. 6, this method has the type 6. See Table 10 and Fig. 10 for examples.

3.4 Method T3Ms

Suppose $p \in \{0, 1,\ldots, n\}$ and $i_1 \in \{0, 1,\ldots, 2n\}$ are parameters and T_1 is a t-norm defined on $[0, i_1]$. The following function defined on $L = \{0, 1,\ldots, n\}$ is a t-norm:

$$T(x,y) = \begin{cases} T_1(x,y) & \text{if } x, y < \min(i_1, p) \\ \min(x,y) & \text{otherwise} \end{cases}.$$

Method T3Ms is based on the ordinal sum of t-subnorms [27] and also referred to as a method of constructing t-norms with shifted domains, see Proposition 6 in [57]. Letter "s" in the name of the method T3Ms denotes "shifted." Figure 4 depicts the location of basic t-norms used in this method in segments of $L \times L$ when $i_1 > p$. When $i_1 < p$, this method can be obtained from the method TMMT when the parameter p is replaced by parameter i_1 and instead of T_4 it is used T_M. See Table 10 and Fig. 10 for examples of such t-norms used as a part of other types of t-norms 3DTs and TMMTs.

Note that the domains of t-norms T_1 used in monotone sum of t-norms (method 4T), in ordinal sum of t-norms (method TMMT) and in T3Ms method generally are different. In the monotone sum of t-norms, T_1 is defined on

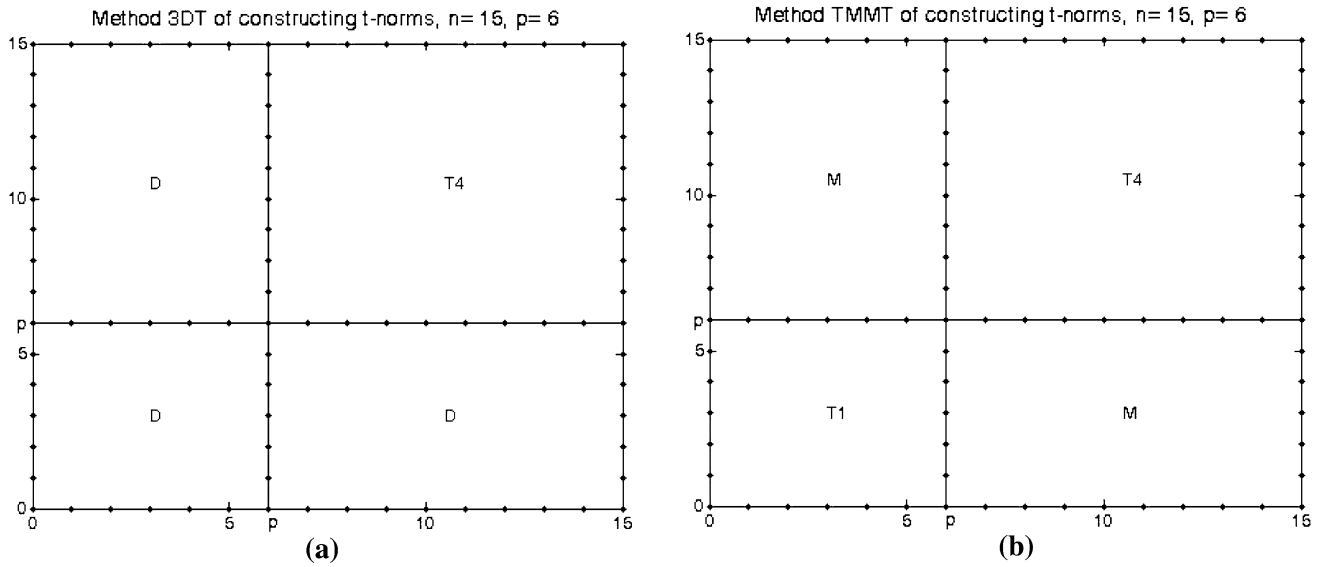

Fig. 3 **a** 3DT and **b** TMMT (ordinal sum) methods of constructing t-norms

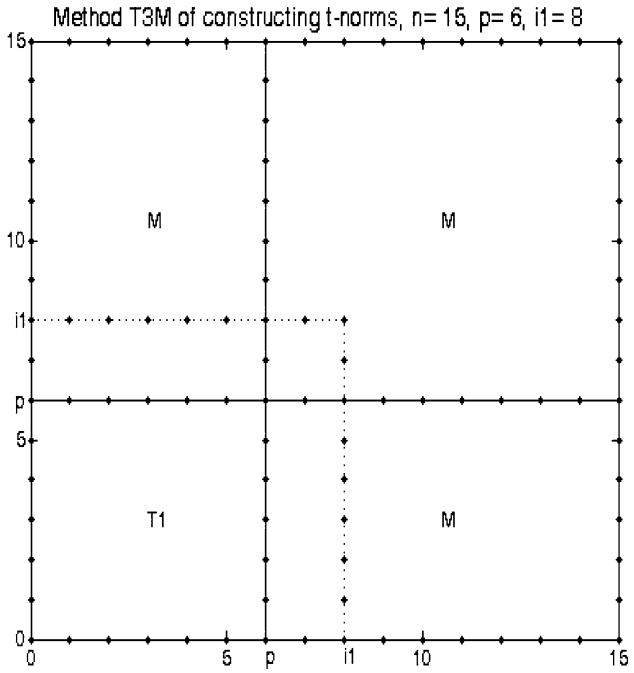

Fig. 4 Method T3Ms. t-Norm T_1 is defined on $[0, i_1]$ and it is "visible" only in "window" $[0, p-1] \times [0, p-1]$

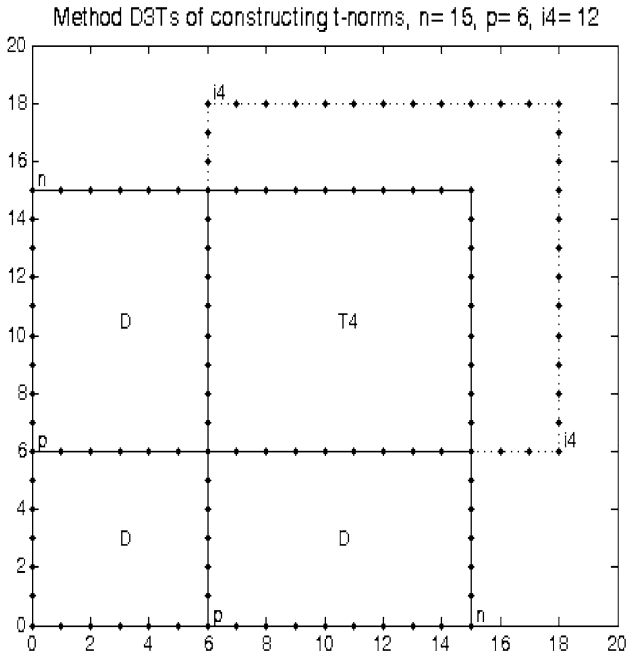

Fig. 5 Method 3DTs. t-Norm T_4 is defined on $[0, i_4]$, where $i_4 = 12$. It is "visible" only in window $[p, n-1] \times [p, n-1] = [6, 14] \times [6, 14]$

$L = \{0, 1,\ldots, n\}$. In the ordinal sum of t-norms, T_1 is defined on "window" $L_1 = [0, p] \times [0, p]$, where $p \in \{0, 1,\ldots, n\}$. In T3Ms method, the domain of t-norm T_1 generally differs from the domains of both of these methods and can be defined on any interval $[0, i_1]$. From this point of view, the domain of T_1 in T3Ms method is "shifted" with respect to the domains of T_1 in the first two methods. For simplicity of hardware implementation, we selected $i_1 \in \{0, 1,\ldots, 2n\}$.

In Table 1 and Fig. 6, this method is used as a part of the methods 3DTs and TMMTs with types 5 and 7 correspondingly when t-norms T_1 and T_4 have shifted domains. These methods are defined as follows.

3.5 Method 3DTs

Suppose T_D is defined on $L = \{0, 1,\ldots, n\}$, $p \in \{0, 1,\ldots, n\}$, $i_4 \in \{0, 1,\ldots, 2n\}$ and t-norm T_4 is defined on $L_4 = \{0, 1,\ldots, i_4\}$. Then for all $x, y \in L = \{0,\ldots, n\}$, the following function is a t-norm on L:

$$T(x,y) = \begin{cases} p + T_4(x-p, y-p) & \text{if } p \le x,y < \min(n, p+i_4) \\ \min(x,y) & \text{if } p+i_4 \le x,y \le n \\ T_D(x,y) & \text{otherwise} \end{cases}.$$

Figure 5 depicts the location of basic t-norms used in this method. See Table 10 and Fig. 10 for examples.

3.6 Method TMMTs

Let $p \in \{0, 1,\ldots, n\}$, $i_1 \in \{0, 1,\ldots, 2n\}$, $i_4 \in \{0, 1,\ldots, 2n\}$, t-norm T_1 be a t-norm on $L_1 = \{0, 1,\ldots, i_1\}$ and T_4 be a t-norm on $L_4 = \{0, 1,\ldots, i_4\}$. Then, the following function is a t-norm on $L = \{0, 1, 2,\ldots, n\}$:

$$T(x,y) = \begin{cases} T_1(x,y), & \text{if } x,y < \min(i_1, p) \\ p + T_4(x-p, y-p), & \text{if } p \le x,y < \min(p+i_4, n) \\ \min(x,y), & \text{otherwise} \end{cases}.$$

Note that when $i_1 < p$, the parameter p is replaced by parameter i_1. See Table 10 and Fig. 10 for examples.

Table 1 gives the type codes of the methods considered above and Fig. 6 depicts the flow diagram of the process of selection of the methods of construction of conjunctors and t-norms based on these type codes in FPGA implementation.

The considered above unification of the simplified versions of different methods discussed in [57] gives the basis for the generation of a variety of parametric families of digital t-norms and conjunctors that have efficient implementation in hardware when we use the simplest t-norms as basic modules. Generally, as in the continuous case [26, 27], the ordinal sum can be applied to more than two summands. Another way to extend the number of diagonal sections is to apply recursively ordinal sum with two summands several times.

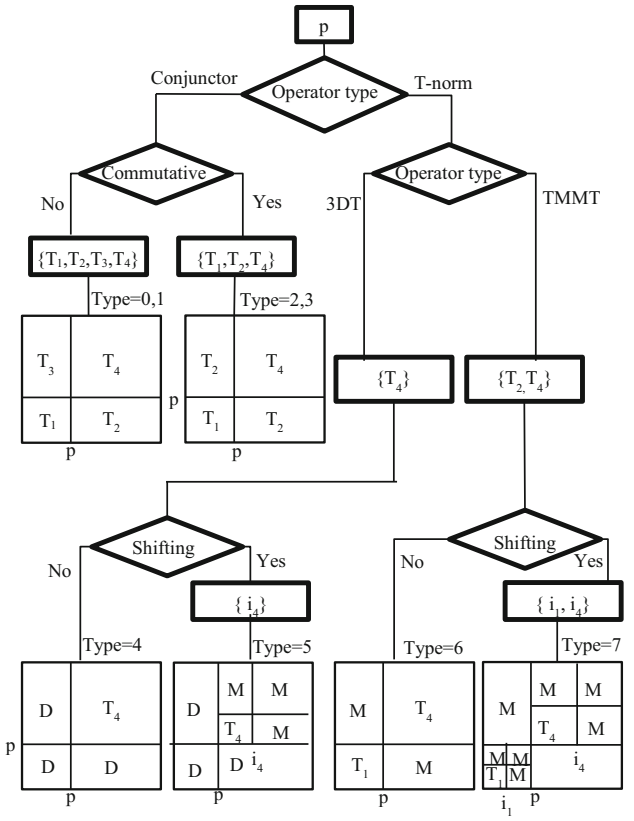

Fig. 6 The flow diagram of the process of selection of operations in FPGA implementation

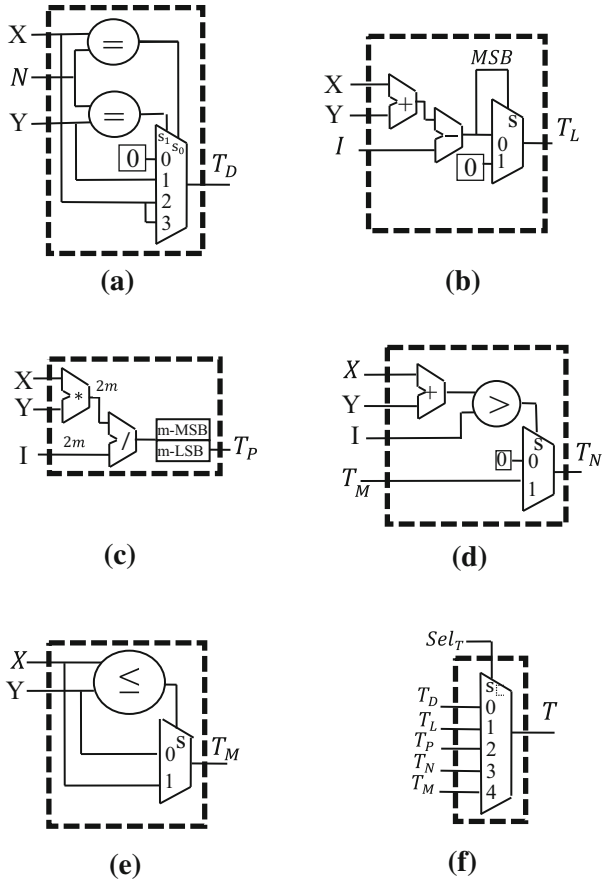

Fig. 7 Logic schemes of basic t-norms: **a** Drastic, **b** Lukasiewicz, **c** Product, **d** Nilpotent, **e** Minimum, **f** t-norms multiplexed

4 Hardware Design of Parametric Operations

In this section, we discuss the design and the implementation on a FPGA of parametric digital conjunctors and t-norms considered in the previous section. First, we show the logic schemes of the implementation of basic t-norms, and after that it will show the methodology to build parametric conjunctors and t-norms in one building block.

4.1 Basic t-Norms Implementation

Figure 7 depicts the schemes of basic t-norms (digital Product is a conjunctor): (a) Drastic, (b) Lukasiewicz, (c) Product, (d) Nilpotent, and (e) Minimum that have one more input unlike the implementation of these t-norms in [47, 53] (with exception of Minimum). The I input is used for specifying parametric domain of a t-norm, where if one t-norm has m-bits representation, the values to I input belong to $[0, 2^{m+1} - 1]$ interval. It also can be observed that for these implementations, we used only simple digital combinatorial circuits as adders, comparators, multiplexors, and multipliers (which can be implemented by using both adder and multiplexor or defined by multiplier embedded on chip).

Because the fact that definition of basic digital t-norms is valid in the interval [0, n] (instead of unit interval), some modifications in their definitions are done.

(i) Due to I input size is defined on $(m + 1)$-bits, the operations (combinatorial circuits) into each module should be defined on $(m + 1)$-bits, in spite of X, Y inputs and T_i output are defined on m-bits.

(ii) $(m + 1)$-bits operations avoid overflow in Nilpotent and Lukasiewicz t-norms $(X + Y)$.

(iii) Special implementation is done for Lukasiewicz t-norm for avoiding negative values. When $X + Y < \mathbf{I}$ (corresponding a negative number in traditional definition), the most significant bit (MSB) bit on the subtract operation of Fig. 7b is set to 1, and these results can be considered as incorrect, and for the other hand when $X + Y \geq \mathbf{I}$, MSB on subtract operation is set to 0. Then, the MSB bit is used as input selector of a multiplexor with both $X + Y - \mathbf{I}$ or 0 inputs, and therefore only correct result will be chosen.

(iv) Due to the multiplication on unsigned integer representation is an operation from m to 2 m-bits,

the result of this operation will be normalized to m-bits. Then, the definition of T_{P} t-norm is modified to

$$T_P(x, y) = x * y/I.$$

With this normalization, all results are assured to be in L, and T_{P} output takes the m-LSBs of the divisor. The disadvantage of this modification for one hand is the aggregation of a divisor, see Fig. 7c. As a consequence, it is more expensive on resource consumed and latency time slower. On the other hand, digital multiplication operation does not satisfy association property due to the discretization of the results, and therefore, this operator is not a t-norm but it is a conjunctor. It will be referred to as a quasi-t-norm. For these reasons, we present in the next section the results of the implementation of parametric operators including and not including product t-norm as a basic t-norm separately.

For getting a conjunctor or t-norm generated from basic t-norm, we multiplexed the different modules as shown in Fig. 7f using a 4-input multiplexor, i.e., a mu $\times$ 4 $\times$ 1 (or mu $\times$ 5 $\times$ 1, when t-product is included), which can be configured as shown in Table 2a (or Table 2b, when t-product is included).

4.2 Implementation of Parametric Conjunctors and t-Norms

For designing parametric conjunctors and t-norms, it is needed to define some input parameters that can be classified as tuning parameters and configuration parameters. The tuning parameters are p, i_1, and i_2, which can take different values in the definition of one conjunctor or t-norm. The configuration parameters T_i are used for specifying conjunctors or t-norms applied in specific method defined by *type* configuration parameter. In spite of the parametric operator has eight input parameters, all of them are not always used to define a specific t-norm or conjunctor. Table 3a shows the input parameters that are used in specific method and Table 3b gives information about the number of bits used by each inputs. *Type* input is not included in Table 3a, but it is clear that this input is necessary in general scheme. The inputs X and Y and Z output are not included in Table, because they are not parameter, but these should be considered as the input and output in the design with m-bits.

Figure 8 shows the implementation schemes of the six different methods. Taken into account, the features of these schemes are as follows: (a) All schemes have different sets of inputs. (b) All implementations need a control module, called ***Ctrl1***. This module is used for splitting the input space LxL in four regions D_i as it is specified in Table 4, and it uses the simple comparators for its implementation. (c) The implementation of a parametric t-norm requires two subtractors and one adder used to compute t-norm specified on region D_4. Therefore, 2-input multiplexors are implemented to choose the inputs and the output of t-norm module, which are controlled by the ANDed output of ***Ctrl1*** module, referred as *xyGEp* (X and Y are greater or equal to p). (d) For implementation of a t-norm with shifted domain, a control module was created, called ***Ctrl2***, which configures the selector of t-norm module, according to the values of D_i, X, Y, I, and T_i. Two control modules are used to reduce the complexity of the design.

Table 2 Selector of t-norms

(a) With 4 basic t-norm	
T_i	T
0	Drastic
1	Lukasiewicz
2	Nilpotent
3	Minimum
(b) With 5 basic t-norm	
T_i	T
0	Drastic
1	Lukasiewicz
2	Product
3	Nilpotent
4	Minimum

Table 3 Input parameters of conjunctors and t-norms methodologies

(a) Input parameters classified by families

Methodology	Tuning parameter	Configuration parameter
4T	p	T_4, T_3, T_2, T_1
4Tc	p	T_4, T_2, T_1
3DT	p	T_4
3DTs	p, i_1	T_4
TMMT	p	T_4, T_1
TMMTs	p, i_1, i_2	T_4, T_1

(b) Number of bits of input parameters

Input parameter	Number of bits
p	m
i_1, i_2	$m+1$
T_i	2 (or 3)
type	3

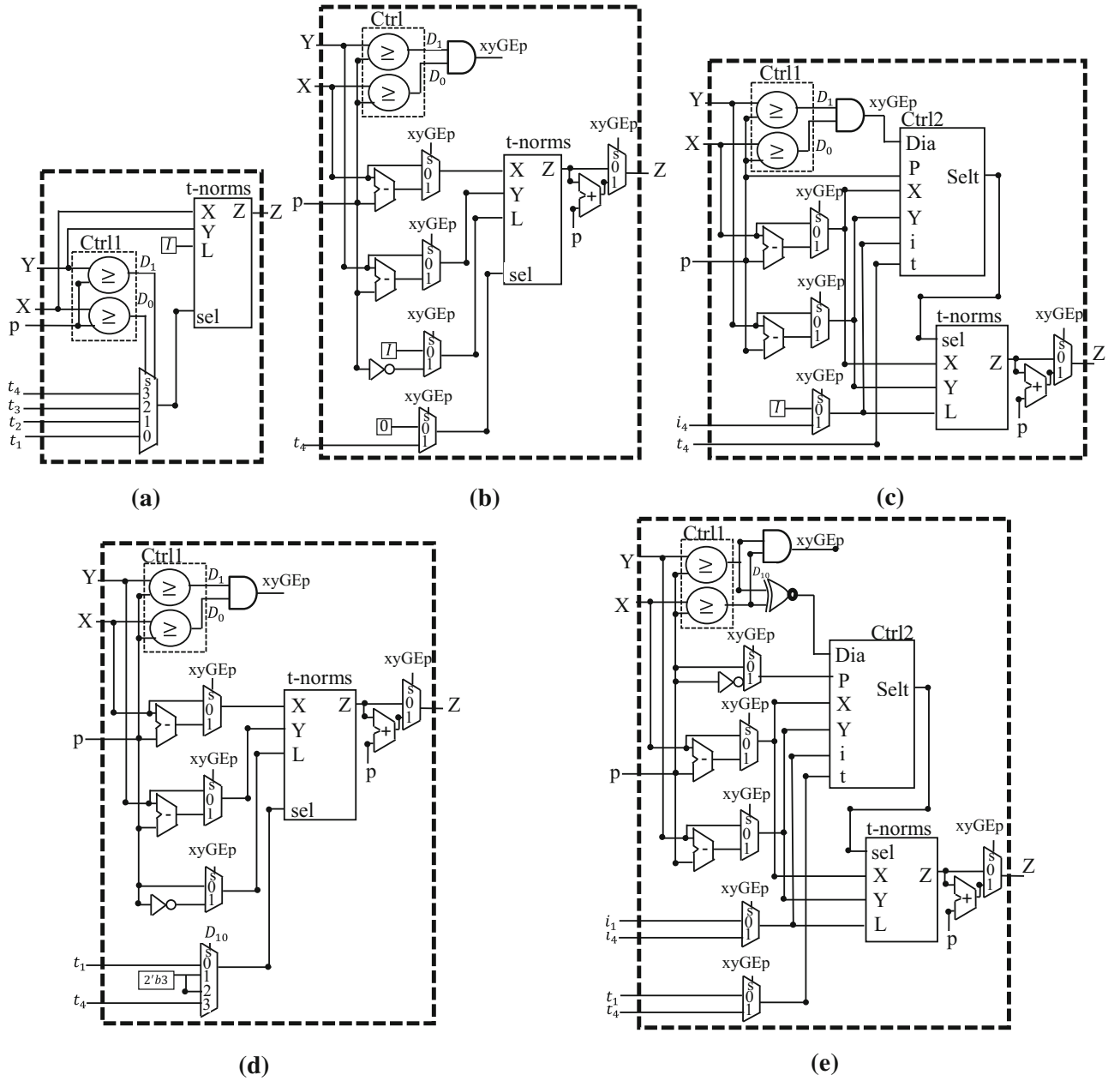

Fig. 8 Schemes of families of parametric conjunctors and t-norms. **a** 4T y 4Tc, **b** 3DT, **c** 3DTs, **d** TMMT, **e** TMMTs

Table 4 Conditions for configuring Ctrl1 module

Inputs	D_i
$x < p$ & $y < p$	D_1
$x < p$ & $y \geq p$	D_2
$x \geq p$ & $y < p$	D_3
$x \geq p$ & $y \geq p$	D_4

In the rows 5 and 7 of Table 5, there are specified the conditions to configure 3DTs and TMMTs methods. In the next section, the latency times and resources used by each implementation in comparison with the general implementation of the operator including all parametric families are shown and analyzed.

4.3 Implementation of the General Operator

To implement the general operator, we use common parts from the different families' implementations as shown in Fig. 8, together with the control module, ***Ctrl***, that generalizes the differences of all types (see Fig. 9). ***Ctrl*** module has *X, Y, D_i, i_i, t_i* and *type* inputs for configuring the sel_t and sel_l and *xyGEP* outputs that are plugged to inputs of t-norm module

Table 5 Conditions for configuring Ctrl module

Type	$x, y \in D_1$			$x, y \in D_2$			$x, y \in D_3$			$x, y \in D_4$		
	sel_T	sel_L	*xyGEp*	sel_T	sel_L	*xyGEp*	sel_T	sel_L	*xyGEp*	sel_T	sel_L	*xyGEp*
0, 1	T_1	*n*	0	T_2	*N*	0	T_3	*n*	0	T_4	*n*	0
2, 3	T_1	*n*	0	T_2	*N*	0	T_2	*n*	0	T_4	*n*	0
4	T_D	*n*	0	T_D	*N*	0	T_D	*n*	0	T_4	*n-p*	*1*
5	T_D	*n*	0	T_D	*N*	0	T_D	*n*	0	$[T_4\|T_M]$**	i_4	*1*
6	T_1	*p*	0	T_M	*N*	0	T_M	*n*	0	T_4	*n-p*	*1*
7	T_1	i_1	0	$[T_1\|T_M]$*	*N*	0	T_M	*n*	0	$[T_4\|T_M]$**	i_4	*1*

* First t-norm is set when *x*, *y* inputs are less than min (*p*, ii), T_M is used otherwise

** T4 t-norm is set when $x - p$, $y - p$ inputs are less than min ($I - p$, *i*4), T_M is used otherwise

and multiplexors according to specification of Table 5. The *p* input parameter is used for dividing the $L \times L$ space on four sections: D_1, D_2, D_3, and D_4 as shown in Fig. 2a. The values of *p* belong to [0, n] interval. *xyGEp* signal is set to 1, when (*x*,*y*) tuple is in the region D_4 and *Type input* is >3, otherwise *xyGEp* signal is set to zero. The input selectors of t-norm and multiplexor MUL_L modules will be configured according to the type of parametric family of t-norm or conjunctors and to some of the input values of the parameters *i*1, *i*4, *X*, *Y*, *p*, corresponding to the type of the operator.

For simplicity of the design, the Ctrl module was implemented using behavioral description, i.e., the behavior of the logic as shown in Table 5 was described using switch-case and if-else Verilog sentences [16].

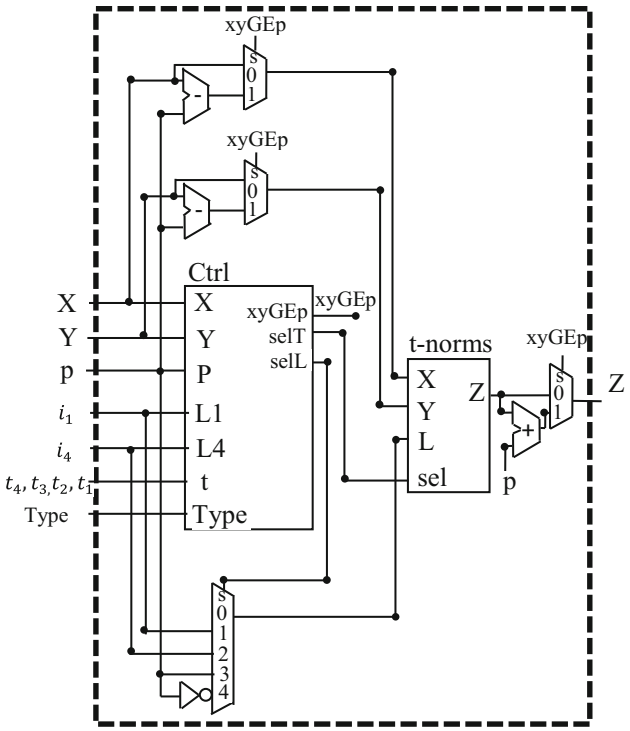

Fig. 9 Logic scheme of the implementation of parametric conjunctors and t-norms

Note that the implementation of the conjunctor is not validated to avoid the user errors, i.e., to verify the fulfillment of the conditions $T_1 \leq T_2 \leq T_4$ and $T_1 \leq T_3 \leq T_4$. Such module is not hard to develop. However, the authors consider that it affects the latency time of the operator, and as it is only necessary for families of conjunctors, so it is not included in the design. The generation of valid conjunctors is the responsibility of the users.

The assignation of specific t-norm to each section is defined by user or, by default, according to the type of the chosen conjunctor or t-norm. The users can define a specific t-norm using T_1, T_2, T_3, T_4 configuration parameters, which can be configured according to the values of Table 2a, b (depending of the number of the basic t-norms), for example, if the user wants to define Lukasiewicz and Nilpotent t-norms in sections D_1 and D_4, respectively, then $T_1 = 1$ and $T_4 = 2$.

When a commutative conjunctor from 4Tc family is selected, the *type* input should be assigned the value 2 or 3, then the value of T_3 input is not matter, because it is set equal to T_2 by default. For the cases of *type* = 4, 5 t-norms, only the value of T_4 input is user-defined and the other ones are set to T_D by default. Finally, for *type* = 6, 7 t-norms, the values of T_2, T_3 are not matter and they are equal to T_M, i.e., sel_T signal is set to 3 by default.

As it was mentioned in Sect. 3, *type* 5 and 7 t-norms can have shifted domains; therefore, the design has $(m + 1)$-bits inputs i_1 and i_4, for setting the domains of T_1 (only for type 7) and T_4, respectively.

In the next section, we will depict some examples of the input configuration for performing some conjunctors and t-norms. Also, the resources and latency time used for the implementation of the operators with different numbers of bits representation are given.

5 Results

The design of parametric operator for hardware implementation on FPGA was implemented on EP4CE115F29C7 device of Cyclone IV E of Altera [71–73], using Verilog

language [16], for four different sizes of digital representations, when m equals to 5, 8, 16, and 32-bits, using product quasi-t-norm (the scheme presented in section IV) and without product quasi-t-norm, respectively.

We present a comparative analysis of the latency time and resources used in the implementations on EP4CE115F29C7 device of (a) the basic t-norms, (b) the six methodologies separately, and (c) the general parametric operator. Table 6 shows the resources and latency times used on the implementation of basics t-norm, in which one can note that all t-norms with exception of product t-norm have similar latency times and number of resources used for their respective implementations. The product t-norm is about 40 times slower in latency time and requires 10 times more resources than other ones, so we recommend including this t-norm in general parametric operator only when it is really necessary.

Table 6 also shows the resources and latency time of t-norm multiplexed and we can see that the time latency is about 14 % slower than for the slowest individual t-norm, and the resources used are less than for the sum of individual t-norms. It is possible because of more combinational functions could be mapped in the same reconfigurable area (Logic Elements) of the device.

Table 7 shows the resources and latency times used in the implementation of the different families of the operations, corresponding to the designs shown in Fig. 8, on the EP4CE115F29C7 device, where basic t-norm product is not included. We can watch that all families have similar latency time for the same m-bit size with the exception of the generation of 4T and 4Tc conjunctors (which are basically implemented only using a multiplexor).

We can say that latency time in parametric conjunctors 4T and 4Tc is about 30 % slower than the slowest individual t-norm (T_L, Lukasiewicz, as shown in Table 6) and they use less resources than the overall sum of individual basic t-norms by the same reason of t-norm multiplexed, and the implementation of a parametric t-norms (3DT, 3DTs, TMMT, and TMMTs) uses about three times more resources than implementation of a parametric conjunctors, and its latency time is about two times slower.

Table 8 shows the values of resources consumed and latency time in the implementation of general operator using and not using product basic t-norm. Without product t-norm, the resources used are less than 1 % of total resources of this device [51] and the results are getting in one clock cycle with $F_{\max} = 60.42$ MHz in the case of 32-bits representation.

Table 6 Cost for implementing basic t-norms

t-Norm or conjunctor	m-Bit	Combinational functions	$F_{\max}$ (MHz)	Latency time (ns)
TM (Minimum)	5	9	337.38	2.96
	8	16	307.69	3.25
	16	32	241.66	4.13
	32	64	191.86	5.21
TN (Nilpotent)	5	17	317.66	3.15
	8	25	300.48	3.33
	16	49	232.72	4.30
	32	97	182.65	5.47
TL (Lukasiewicz)	5	17	268.89	3.72
	8	26	267.31	3.74
	16	50	217.39	4.60
	32	98	170.33	5.87
TD (Drastic)	5	9	477.78	2.09
	8	14	415.11	2.41
	16	27	377.07	2.65
	32	54	252.53	3.96
TP (Product)	5	111 (1)	46.9	21.32
	8	279 (1)	23.47	42.61
	16	1079 (2)	8.7	114.94
	32	4296 (8)	2.91	343.64
Multiplexed without product	5	35	234.74	4.26
	8	54	201.41	4.96
	16	99	178.67	5.60
	32	189	148.68	6.73

Information in brackets indicates the number of embedded multipliers used

Table 7 Cost for implementing parametric conjunctors and t-norm families

t-Norm or conjunctor	m-Bits	Combinational functions	F_{max} (MHz)	Latency (ns)
4T, 4Tc	5	46	192.53	5.19
	8	74	190.77	5.24
	16	134	164.28	6.09
	32	257	128.83	7.76
3DT	5	91	113.52	8.81
	8	141	98.02	10.20
	16	272	76.81	13.02
	32	533	59.42	16.83
3DTs	5	115	106.87	9.36
	8	180	98.99	10.10
	16	344	78.2	12.79
	32	668	60.64	16.49
TMMT	5	92	105.21	9.50
	8	141	105.57	9.47
	16	272	76.64	13.05
	32	533	62.10	16.10
TMMTs	5	129	110.38	9.06
	8	202	98.91	10.11
	16	381	73.13	13.67
	32	739	63.67	15.71

Comparing this result with the implementation of separated t-norm and conjunctors families, we can determine that the latency time is similar, and resources used are only 14 % more than the family TMMTs, so we can say that there is no degradation in latency time, and cost of resources is good for implementing general parametric operator. These results can be attributed to similar structures that have the six different families of t-norm and conjunctors, and optimization performed by Altera synthesizer, and mapped of different functions in same reconfigurable area.

Table 8 also shows latency time and resources used, when product t-norm is included, and as we expected, the cost of the operator is very bigger compared with same operator without the product t-norm. However, its implementation is feasible because this uses 4 % of total resources of the device, and its latency time is in the order of nanoseconds.

Finally, we present a comparison of our results with the results presented in [31] in which the cost of hardware implementation of one conjunctor was reported. In that work, the implementation was performed on 3E3S500 EFG320-5 FPGA from Xilinx and the parametric conjunctor used 382 gates and latency time of 14.03 ns on 8-bits representation. For better comparison, we reimplement the operator on Altera FPGA using basic t-norm introduced in Fig. 7. Table 9 presents the cost of implementation of the parametric conjunctor and it also presents

Table 8 Resources and latency time used for implementing parametric operator

Category resource	With product quasi-t-norm				Without product				Total on FPGA
	$n = 5$	$n = 8$	$n = 16$	$n = 32$	$n = 5$	$n = 8$	$n = 16$	$n = 32$	
Total combinational functions	242 (0.2 %)	467 (0.4 %)	1424 (1.2 %)	5068 (4.4 %)	162 (0.1 %)	239 (0.2 %)	442 (0.4 %)	849 (0.7 %)	114,480
Embedded multiplier 9-bit elements	1 (0.2 %)	1 (0.2 %)	2 (0.4 %)	8 (1.5 %)	0	0	0	0	532
F_{max} (MHz)	35.3	19.78	7.99	2.79	102.27	75.78	75.78	60.42	
Latency time (ns)	28.33	50.56	125.16	358.42	9.52	10.94	13.90	24.24	
PIN	40	68	116	212	46	64	112	400	529

Table 9 Cost for implementing individual parametric conjunctor and t-norm

t-Norm or conjunctor	Family	Bits	Combinational functions	F_{max} (MHz)	Latency time (ns)
Operator from [31]		5	50	134.26	7.45
		8	82	121.97	8.20
		16	162	90.63	11.03
		32	322	68.67	14.56
Lukasiewicz	3DT	5	80	129.99	7.69
		8	121	114.82	8.71
		16	239	82.67	12.10
		32	462	65.92	15.17
Nilpotent	3DTs	5	111	115.17	8.68
		8	177	107.09	9.34
		16	340	80.76	12.38
		32	673	61.9	16.16
Lukasiewicz-Nilpotent	TMMT	5	81	120.98	8.27
		8	126	108.67	9.20
		16	246	83.79	11.93
		32	486	65.56	15.25
Drastic-Lukasiewicz	TMMTs	5	107	138.2	7.24
		8	156	115.19	8.68
		16	302	98.67	10.13
		32	596	74.34	13.45
LLLM	4Tc	5	36	207.51	4.82
		8	59	196.46	5.09
		16	115	179.82	5.56
		32	451	135.78	7.36

the cost of some of the conjunctors and t-norm designed from a particular family introduced in this work. From this table, we can infer that the conjunctor generated from 4T and 4Tc families is the fastest and with low-complexity of resources and the conjunctor from [31] has similar latency times that of t-norms presented in this work.

We did not implement traditional complex parametric t-norm such as Yager, Hamacher, Dombi, etc., analyzed in [4, 48] because we can deduce that these do not have efficient hardware implementation due to a) these require product t-norm and other complex mathematic functions and b) their implementation is not good for digital representation.

For testing, the hardware implementation was simulated in model-sim simulator of mentor graphics using a 5-bits representation ($N \in [0-31]$), then the design was synthesized in the FPGA device and for verifying its hardware operation, it used hardware in the loop (HIL) approach, in which basically all possible values of (X, Y) from Matlab software to FPGA implementation were send, and the values of Z output of FPGA were returned to Matlab, for more details of HIL see [57, 58], and we obtained same results that in simulations. The data obtained were potted using Matlab software. For other representations, n equal to 8, 16, and 32, only some samples were verified.

Figure 10 shows the graphic surfaces corresponding to the t-norms and conjunctors specified in Table 10 for different values of parameterized inputs. Note that the cells with X value, corresponds to do not care condition, and then their values are not considered. With Fig. 10 and Table 10, we are trying to show all general different cases of configuring the operator because it is not practical to show each different family of the operator (see Table 11).

Note that the cells with X value correspond to do not care condition, and then their values are not considered. With Fig. 10 and Table 10, we are trying to show all general different cases of configuring the operator because it is not practical to show each different family of the operator (see Table 11).

Finally, Table 11 shows the number of conjunctors and t-norms families that can be configured according to number of basic t-norms used, i.e., for 3 (as was implemented in [53]), 4 and 5 basic t-norms (presented in this paper). The cases of commutative and non-commutative

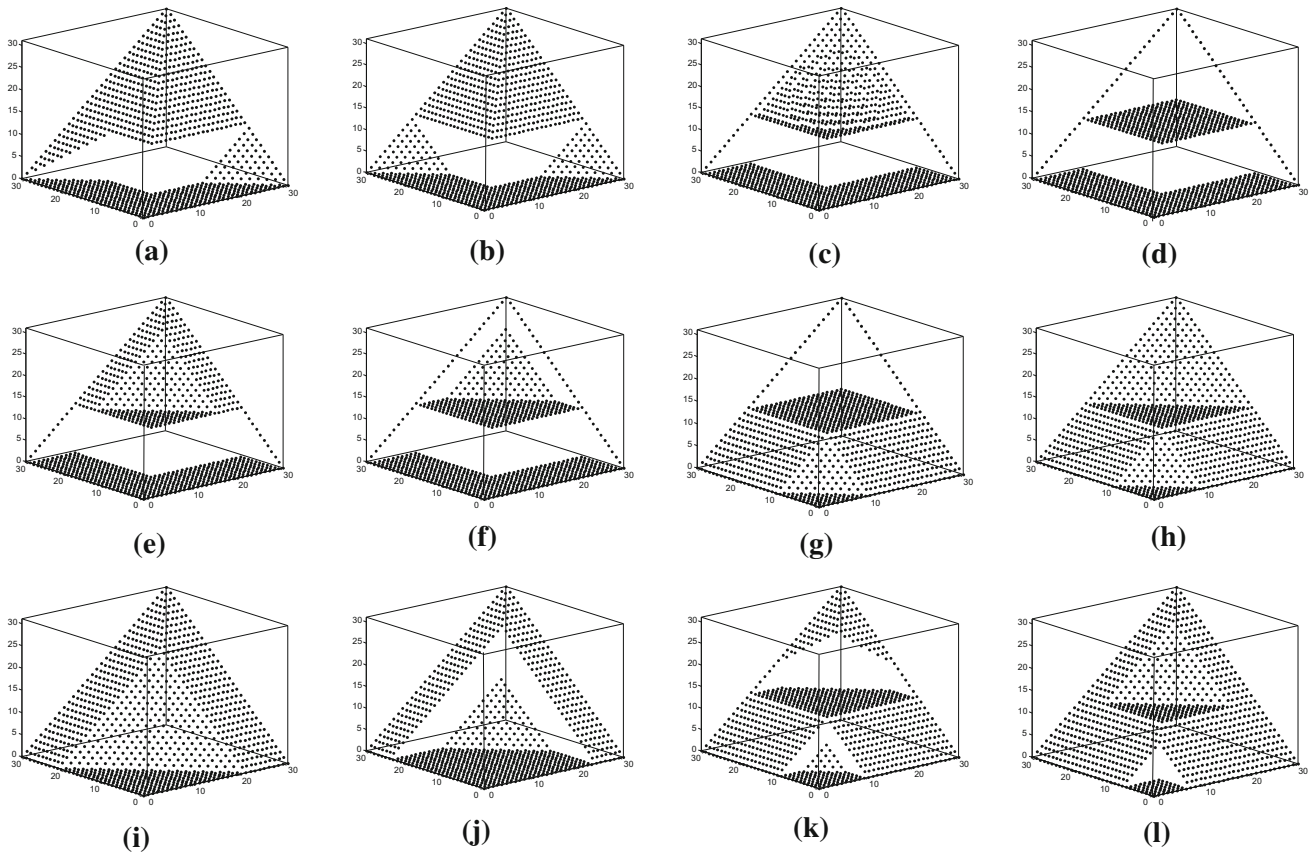

Fig. 10 Surfaces of t-norms and conjunctors corresponding to Table 10

Table 10 Configuration of t-norms and conjunctors

Surface in Fig. 10	Type	Methodology	Basic t-norms used	sel_i (*octal*)	P	i_1	i_4
(a)	0, 1	4T	$T_1 = T_D,\ T_2 = T_L,\ T_3 = T_N,\ T_4 = T_M$	0134	11	X	X
(b)	2, 3	4Tc	$T_1 = T_D,\ T_2 = T_L,\ T_4 = T_M$	01X4	10	X	X
(c)	4	3DT	$T_4 = T_P$	0002	11	X	X
(d)	4	3DT	$T_4 = T_D$	XXX0	11	X	X
(e)	5	3DTs	$T_4 = T_L$	XXX0	11	X	15
(f)	5	3DTs	$T_4 = T_L$	XXX0	11	X	25
(g)	6	TMMT	$T_1 = T_L,\ T_4 = T_D$	1XX0	11	X	X
(h)	6	TMMT	$T_1 = T_L,\ T_4 = T_L$	1XX1	11	X	X
(i)	7	TMMTs	$T_1 = T_L,\ T_4 = X$	1XXX	31	22	X
(j)	7	TMMTs	$T_1 = T_L,\ T_4 = T_M$	1XX4	25	35	x
(k)	7	TMMTs	$T_1 = T_L,\ T_4 = T_N$	1443	11	15	25
(l)	7	TMMTs	$T_1 = T_D,\ T_4 = T_L$	1XX3	11	18	15

conjunctors using 5 basic t-norm were taking into account that T_P and T_N cannot be compared, and so both of them cannot be included in the same parametric conjunctor. Therefore, the total different parametric operators that can be implemented in general operator are 41, 95, or 135 when 3, 4, or 5 different basic t-norms are used, respectively.

Table 11 Classification of t-norms by type input

Type methodology	Number of operators using K basic t-norms		
	$K = 3$	$K = 4$	$K = 5$
Non-commutative conjunctor	10	30	50
Commutative conjunctor	7	16	25
TMMT	9	16	25
TMMTs	9	25	25
3DT	3	4	5
3DTs	3	4	5
Total operators	41	95	135

6 Conclusions

This paper presents the implementation of a generalized digital operator on FPGA that can be configured as a parametric conjunctor or a t-norm in fuzzy inference system. This design consists of Drastic, Lukasiewicz, Product, Nilpotent, and Minimum basic t-norms. This design is more complicated than the another one presented in [53], because it includes more basic t-norms, different methods of aggregation of t-norms such as an ordinal sum of t-norms, an ordinal sum of subnorms based on basic t-norms, monotone sum of t-norms, and extension of basic t-norms by Drastic t-norm [28, 47–50]. The efficient implementation in one scheme of the several parametric methods of generation of conjunctors and t-norms was achieved due to a unified representation of simplified versions of different methods of aggregation of t-norms considered in [47].

This design has efficient implementation with similar latency time of the different families presented, and the resources used are less than 1 % of the total resources of EP4CE115F29C7 device FPGA.

This design can be extended directly for implementation also $(p, I - p)$-monotone sum methodology presented in [54] by a simple modification in control unit. We present the results of this operator implementation considering until a 32-bits for digital representation, because we think that for a more precise representation it is better to use floating point representation.

This implementation does not contain a validation module for protecting operator from bad configurations that user can generate, because such module will make the implementation slower. For solving this problem, we recommend two solutions: (a) to make the validation in software or (b) to make a hardware validation module as it was done in [53].

As it was shown in Sect. 5, the implementation can be configured to obtain one of the 135 operators; therefore, we can suppose that this operator is very useful at implementations of adaptive fuzzy systems with reconfigurable logic operations that include a learning algorithm.

Acknowledgments The work was partially supported by the project 20151589 of Instituto Politécnico Nacional, DF, Mexico.

References

1. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
2. Yen, J., Langari, R., Zadeh, L.A.: Industrial Applications of Fuzzy Logic and Intelligent Systems. IEEE Press, Piscataway (1995)
3. Kacprzyk, J., Pedrycz, W. (eds.): Springer Handbook of Computational Intelligence. Springer, Berlin (2015)
4. Kandel, A., Langholz, G.: Fuzzy Hardware: Architectures and Applications. Kluwer Academic Publishers, Dordrecht (1997)
5. Liang, Y.W., Xu, S.D., Liaw, D.C., Chen, C.C.: A study of T-S model-based SMC scheme with application to robot control. Ind. Electron. IEEE Trans. **55**(11), 3964–3971 (2008)
6. El-Nagar, A.M., El-Bardini, M.: Derivation and stability analysis of the analytical structures of the interval type-2 fuzzy PID controller. Appl. Soft Comput. **24**, 704–716 (2014)
7. Bosque, G., Del Campo, I., Echanobe, J.: Fuzzy systems, neural networks and neuro-fuzzy systems: a vision on their hardware implementation and platforms over two decades. Eng. Appl. Artif. Intell. **32**, 283–331 (2014)
8. Govindasamy, K., Neeli, S., Wilamowski, B.M.: Fuzzy system with increased accuracy suitable for FPGA implementation. In: INES 2008, Intelligent Engineering Systems Conference, pp. 133–138. (2008)
9. Xue, J., Sun, L., Liu, M., Qiao, C., Ye, G.: Research on high-speed fuzzy reasoning with FPGA for fault diagnosis expert system. In: Proceedings of the International Conference on Mechatronics and Automation, pp. 3005–3009. (2009)
10. Brox, M., Sánchez-Solano, S., del Toro, E., Brox, P., Moreno-Velo, F.J.: CAD tools for hardware implementation of embedded fuzzy systems on FPGAs. Ind. Inform. IEEE Trans. **9**(3), 1635–1644 (2013)
11. Khalaji, A., Seyedtabaii, S.: Intricacies in digital CMOS implementation of a reconfigurable fuzzy logic traffic light controller. In: 5th conference on information and knowledge technology (IKT), pp. 187–191. (2013)
12. Seyedtabaii, S., Khalaji, A.: Single chip digital CMOS implementation of a reconfigurable fuzzy logic traffic controller. J. Intell. Fuzzy Syst. **27**(2), 921–928 (2014)
13. Rashidi, B., Sayedi, S. M.: A high-speed multiplexer-based fine-grain pipelined architecture for digital fuzzy logic controllers. Int. J. Electron., 1–19 (2015). **(in print)**
14. http://www.xilinx.com/fpga/
15. Kilts, S.: Advanced FPGA. Design Architecture, Implementation, and Optimization. Wiley, Hoboken (2007)
16. Palnitkar, S.: Verilog HDL: A Guide to Digital Design and Synthesis, 2nd edn. Prentice Hall Professional (2003)

17. Cortés-Antonio, P., González, J.R., Villa-Vargas, L.A., Ramırez-Salinas, M.A., Molina-Lozano, H., Batyrshin, I.: Design and implementation of differential evolution algorithm on FPGA for double-precision floating-point representation. Acta Polytechnica Hungarica **11**(4), 139–153 (2014)
18. Yiu, K.F.C., Li, Z., Low, S.Y., Nordholm, S.: FPGA multi-filter system for speech enhancement via multi-criteria optimization. Appl. Soft Comput. **21**, 533–541 (2014)
19. Sivakumar, L., Devi, S.: Implementation of VLSI model as a tool in diagnostics of slowly varying process parameters which affect the performance of steam turbine. Appl. Soft Comput. **24**, 730–741 (2014)
20. Lopez Leyva, J.A., Ureña Castro, J.L., Sanchez, J.D.D., Mendieta Jimenez, F.J., Arvizu Mondragon, A.: FPGA-based emulation of a synchronous phase-coded quantum cryptography system. Computación y Sistemas **19**(1), 185–195 (2015)
21. Godo, L., Lopez de Mantaras, R., Sierra, C., Verdaguer, A.: Milord: the architecture and management of linguistically expressed uncertainty. Int J. Intell. Syst. **4**, 471–501 (1989)
22. Reyes, J.A., Esteva, F., Puyol-Gruart, J.: Defining and combining multiple-valued logics for knowledge-based systems. Congrès Català d'Intelligència Artificial (1998)
23. Batyrshin, I., Kaynak, O.: Parametric classes of generalized conjunction and disjunction operations for fuzzy modeling. IEEE Trans. Fuzzy Syst. **7**, 586–596 (1999)
24. Batyrshin, I., Kaynak, O., Rudas, I.: Fuzzy modeling based on generalized conjunctor operations. IEEE Trans. Fuzzy Syst. **10**, 678–683 (2002)
25. Klir, G.J., Folger, T.A.: Fuzzy Sets, Uncertainty, and Information. Prentice-Hall, Englewood Cliffs (1988)
26. Klement, E.P., Mesiar, R., Pap, E.: Triangular Norms. Kluwer, Dordrecht (2000)
27. Jenei, S.: How to construct left-continuous triangular norms—state of the art. Fuzzy Sets Syst. **143**, 27–45 (2004)
28. Koprinkova-Hristova, P.D.: Fuzzy operations' parameters versus membership functions' parameters influence on fuzzy control systems properties. In: 2nd IEEE International Conference on Intelligent Systems, pp. 219–224. (2004)
29. Rudas, I.J., Fodor, J.: Information aggregation in intelligent systems using generalized operators. Int. J. Comput. Commun. Control **1**(1), 47–57 (2006)
30. Preitl, S., Precup, R.E., Fodor, J., Bede, B.: Iterative feedback tuning in fuzzy control systems. Theory and applications. Acta Polytechnica Hungarica **3**(3), 81–96 (2006)
31. Zhang, X., He, H., Xu, Y.: A fuzzy logic system based on Schweizer–Sklar t-norm. Sci. China Ser. F Inf. Sci. **49**(2), 175–188 (2006)
32. Beliakov, G., Pradera, A., Calvo, T.: Aggregation Functions: A Guide for Practitioners, vol. 361. Springer, Heidelberg (2007)
33. Alcalá-Fdez, J., Herrera, F., Márquez, F., Peregrín, A.: Increasing fuzzy rules cooperation based on evolutionary adaptive inference systems. Int J. Intell. Syst. **22**, 1035–1064 (2007)
34. Rudas, I.J., Batyrshin, I.Z., Zavala, A.H., Nieto, O.C., Horváth, L., Vargas, L.V.: Generators of Fuzzy Operations for Hardware Implementation of Fuzzy Systems. In MICAI 2008: Advances in Artificial Intelligence, pp. 710–719. Springer, Berlin (2008)
35. Rudas I.J., Batyrshin I.Z., Hernández Zavala A., Camacho Nieto O., Villa Vargas L. Digital fuzzy parametric conjunctions for hardware implementation of fuzzy systems. In: ICCC2009, IEEE 7th International Conference on Computational Cybernetics, November 26–29, 2009, Palma de Mallorca, Spain, pp. 157–166
36. Hernández Zavala, A., Batyrshin, I., Camacho Nieto, O., Rudas, I.J., Sidorov, G., Villa, L.A.: On generation and FPGA implementation of digital fuzzy parametric conjunctions. Appl. Comput. Math. **11**(2), 150–164 (2012)
37. Mayor, G., Torrens, J.: Triangular norms on discrete settings. In: Klement, E.P., Mesiar, R. (eds.) Logical, Algebraic, Analytic and Probabilistic Aspects of Triangular Norms, pp. 189–230. Elsevier, New York (2005)
38. Mayor, G., Monreal, J.: Additive generators of discrete conjunctive aggregation operations. IEEE Trans. Fuzzy Syst. **15**, 1046–1052 (2007)
39. Batyrshin, I.Z., Rudas, I.J., Panova, A.: On generation of digital fuzzy parametric conjunctions. Stud. Comput. Intell. **243**, 79–89 (2009)
40. Korytkowski, M., Scherer, R.: Modular neuro-fuzzy systems based on generalized parametric triangular norms. In: Parallel Processing and Applied Mathematics, pp. 332–339. Springer, Berlin (2010)
41. Drewniak, J., Krol, A.: A survey of weak connectives and the preservation of their properties by aggregations. Fuzzy Sets Syst. **161**(2), 202–215 (2010)
42. Krol, A.: Dependencies between fuzzy conjunctions and implications. In EUSFLAT 2011 Conference, pp. 230–237. (2011)
43. Król, A. Generating of fuzzy implications. In EUSFLAT 2013 Conference, pp. 758–763. (2013)
44. Lovassy, R., Kóczy, L.T., Gál, L.: Function approximation performance of fuzzy neural networks. Acta Polytechnica Hungarica **7**(4), 25–38 (2010)
45. Lovassy, R., Koczy, L.T., Gal, L.: Generalization capability of neural networks based on fuzzy operators. Appl. Comput. Math. **10**(2), 340–355 (2011)
46. Batyrshin, I.: Uncertainties with memory in construction of strict monotonic t-norms and t-conorms for finite ordinal scales: basic definitions and applications. Appl. Comput. Math. **10**(3), 498–513 (2011)
47. Yan, H.B., Huynh, V.N., Nakamori, Y., Murai, T.: On prioritized weighted aggregation in multi-criteria decision making. Expert Syst. Appl. **38**(1), 812–823 (2011)
48. Tellez, A., Molina, H., Villa, L., Rubio, E., Batyrshin, I.: Parametric type-2 fuzzy logic systems. Chapter 5. In: Dadios, E.P. (ed.). Fuzzy Logic—Algorithms, Techniques and Implementations, pp. 97–114. http://www.intechopen.com/books/fuzzy-logic-algorithms-techniques-and-implementations/parametric-type-2-fuzzy-systems (2012)
49. Rybalov, A., Kagan, E., Yager, R.: Parameterized uninorm and absorbing norm and their application for logic design. In: Electrical & Electronics Engineers in Israel (IEEEI), 2012 IEEE 27th Convention, pp. 1–5. IEEE (2012)
50. Aras, A.C., Kaynak, O., Abiyev, R.: Slip control of a quarter car model based on type-1 fuzzy neural system with parameterized conjunctions. In IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society, pp. 2488–2493. IEEE (2012)
51. Aras, A.C., Kaynak, O.: Trajectory tracking of a 2-DOF helicopter system using neuro-fuzzy system with parameterized conjunctors. In Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference, pp. 322–326. IEEE (2014)
52. Aras, A.C., Kaynak, O.: Fuzzy interval TSK type-2 modeling with parameterized conjunctors. Asian J. Control **17**(3), 963–976 (2015)
53. Rak, E.: The distributivity property of increasing binary operations. Fuzzy Sets Syst. **232**, 110–119 (2013)
54. Kagan, E., Rybalov, A., Siegelmann, H., Yager, R.: Probability-generated aggregators. Int. J. Intell. Syst. **28**(7), 709–727 (2013)
55. Yu, X., Xu, Z., Liu, S.: Prioritized multi-criteria decision making based on preference relations. Comput. Ind. Eng. **66**(1), 104–115 (2013)
56. Márquez, A.A., Márquez, F.A., Roldán, A.M., Peregrín, A.: An efficient adaptive fuzzy inference system for complex and high

dimensional regression problems in linguistic fuzzy modelling. Knowl. Based Syst. **54**, 42–52 (2013)
57. Batyrshin, I.Z., Rudas, I.J., Villa, L.A., Cortes-Antonio, P.: On the monotone sum of basic t-norms in the construction of parametric families of digital conjunctors for fuzzy systems with reconfigurable logic. Knowl. Based Syst. **38**, 27–36 (2013)
58. Kóczy, L.T.: A note on Hamacher-operators. In: Advances in Soft Computing, Intelligent Robotics and Control, pp. 159–163. Springer International Publishing, Berlin (2014)
59. Sidorov, G., Gelbukh, A., Gómez-Adorno, H., Pinto, D.: Soft similarity and soft cosine measure: similarity of features in vector space model. Computación y Sistemas **18**(3), 491–504 (2014)
60. Kagan, E., Rybalov, A., Sela, A., Siegelmann, H., Steshenko, J.: Probabilistic control and swarm dynamics in mobile robots and ants. In: Burki, S., Dobbie, G., Koh, Y. (eds.) Biologically-Inspired Techniques for Knowledge Discovery and Data-Mining, pp. 11–47. IGI Global, Hershey (2014)
61. Meschino, G.J., Comas, D.S., Ballarin, V.L., Scandurra, A.G., Passoni, L.I.: Automatic design of interpretable fuzzy predicate systems for clustering using self-organizing maps. Neurocomputing **147**, 47–59 (2015)
62. Bentkowska, U., Król, A.: Preservation of fuzzy relation properties based on fuzzy conjunctions and disjunctions during aggregation process. Fuzzy Sets Syst. (2015). doi:10.1016/j.fss.2015.06.001
63. Batyrshin, I., Hernández, Zavala A., Camacho, Nieto O., Villa Vargas, L.: Generalized Fuzzy Operations for Digital Hardware Implementation, MICAI 2007, LNAI 4827, pp. 9–18. Springer, Berlin (2007)
64. Zavala, A.H., Nieto, O.C., Batyrshin, I., Vargas, L.V.: VLSI implementation of a module for realization of basic t-norms on fuzzy hardware. FUZZ-IEEE 2009, IEEE International Conference on Fuzzy Systems, 20–24 Aug, pp. 655–659. Jeju Island, Korea (2009)
65. Hernández Zavala, A., Batyrshin, I.Z., Rudas, I.J., Villa Vargas, L., Camacho Nieto, O.: Parametric operations for digital hardware implementation of fuzzy systems. In: Hernández Aguirre, A., et al. (eds.): MICAI 2009, Lecture Notes on Artificial Intelligence, vol. 5845, pp. 432–443. Springer, Berlin (2009)
66. Hernández Zavala, A., Batyrshin, I.Z., Camacho Nieto, O., Castillo, O.: Conjunction and disjunction operations for digital fuzzy hardware. Appl. Soft Comput. **13**(7), 3248–3258 (2013)
67. Cortés Antonio, P., Batyrshin, I., Rudas, I., Panova, A., Villa Vargas, L.A.: FPGA implementation of (p)-monotone sum of basic t-norms. In: IEEE World Congress on Computational Intelligence, FUZZ-IEEE, pp. 1491–1497. Barcelona, Spain (2010)
68. Cortés Antonio, P., Batyrshin, I.Z., Villa Vargas, L.A., Rudas, I., Molina Lozano, H.: FPGA Implementation of (p, I-p)-monotone sum of basic t-norms. In: Proceedings of the World Conf. Soft Computing, San Francisco, USA (2011)
69. Cortés Antonio, P., Batyrshin, I., Molina Lozano, H., Villa Vargas, L.A., Rudas, I. FPGA implementation of fuzzy system with parametric membership functions and parametric conjunctions. In: Advances in Soft Computing. Lecture Notes on Artificial Intelligence, vol. 6438, pp. 487–499. MICAI 2010, November 8–13, Pachuca, Mexico. Springer, Berlin (2010)
70. Cortes Antonio, P., Batyrshin, I., Lozano, H.M., Salinas, M.A.R., Vargas, L.V.: FPGA implementation of fuzzy Mamdani system with parametric conjunctions generated by monotone sum of basic t-norms. Polibits **44**, 53–58 (2011)
71. Altera: http://www.altera.com/ (2011)
72. Altera: https://www.altera.com/en_US/pdfs/literature/wp/wp-01208-hardware-in-the-loop.pdf (2013)
73. Altera: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/ug/ug_system_console.pdf (2014)

Prometeo Cortés-Antonio is a research student of Center for Computer Research of the National Polytechnic Institute (IPN), DF, México. He received his M.S. degree in computer science from IPN in 2012. He received the B.S degree in Electronic Engineering from Universidad Tecnológica de la Mixteca, México in 2004. He is interested in and working in the areas of Soft Computing, Evolvable Hardware and Digital Systems.

Ildar Batyrshin received his Diploma from Moscow Physical-Technical Institute, Russia, in 1975; Ph.D. from Moscow Power Engineering Institute in 1983; and Dr. Sci. (Habilitation) degree from Higher Attestation Committee of Russian Federation in 1996. Since 1975 he has been working in the Department of Informatics and Applied Mathematics of Kazan State Technological University, Russia (as a Department Head during 1997–2003). Since 1999, he has also been working at the Institute of Problems of Informatics of Academy of Sciences of the Republic of Tatarstan, Russia, as a leading researcher. From 2003 to 2014, he was working as a leading researcher and a Head of the project at the Mexican Petroleum Institute. He is currently working at the Center for Computing Research of the National Polytechnic Institute of Mexico as a Full-time Professor. His areas of research activity are Soft Computing, Artificial Intelligence, and Data Mining.

Luis A. Villa-Vargas received his B.Sc. degree in Electronic Engineering and M.Sc. degree in Computer Science from National Polytechnic Institute (IPN), Mexico, in 1992 and in 1994, respectively, and his Ph.D. in Computer Science from the Polytechnic University of Catalonia in 1999. From December 1999 to February 2001, he was with the Laboratory for Computer Science as a Postdoctoral Fellow at the Massachusetts Institute of Technology, working in the SCALE project. From October 2001 to January 2007, he was with the Mexican Petroleum Institute. Since January 2007, he has been working with the Center for Computer Research at the National Polytechnic Institute in Mexico. His research interests include high-performance computer microarchitecture, VLSI design for energy-efficient computing, embedded systems, and fuzzy systems. He is the member of the National System of Researchers in Mexico, and currently he is the Director of the Center for Computing Research of National Polytechnic Institute.

Imre J. Rudas received the Master Degree in Mathematics in Budapest and the Doctor of Science degree from the Hungarian Academy of Sciences. He is a Full-time Professor and Head of Óbuda University Research and Innovation Center. His current areas of research activities are Robotics and Computational Intelligence. He has published 6 books and more than 690 scientific papers.

Herón Molina-Lozano received the B. Eng. degree from the Escuela Superior de Ingenieria Mecanica y Electrica (ESIME), campus Zacatenco, Instituto Politecnico Nacional (IPN), Mexico, and the Dr. Sci. degree from the Departamento de Ingenieria Electrica, Centro de Investigacion y de Estudios Avanzados (CINVESTAV) del Instituto Politécnico Nacional, campus Mexico City, in 1991 and 2001, respectively. He also received the Master in Computer Sciences (MCC) from the Instituto Tecnologico y de Estudios Superiores de Monterrey (ITESM), campus Estado de Mexico, Mexico, in 1998. He is currently working as a Full-time Professor in the Microtechnology and Embedded Systems Laboratory, Centro de Investigacion en Computacion (CIC) of the IPN. His research interests and publications are in the areas of Soft Computing, Bioinformatics, Analog CMOS Design, and Digital FPGA Design.

Marco A. Ramírez-Salinas received the B.Sc. degree (1995) from the Superior School of Mechanical and Electrical Engineering (ESIME) and M.Sc. degree (2002) from the Center for Computing Research (CIC), both from the National Polytechnic Institute (IPN) of Mexico, and the Ph.D. degree in Computer Science (2007) from the Department of Computer Architecture of Polytechnic University of Catalonia in Barcelona, Spain. He is currently working as a Full-time Professor in the Department of Computer Engineering of CIC-IPN, Mexico. His current research interests and publications are in the areas of Computer Architecture, Digital System Design, and Sensor Devices.