Int. J. Appl. Math. Comput. Sci., 2013, Vol. 23, No. 2, 473–483
DOI: 10.2478/amcs-2013-0036

DECOMPOSITION OF THE FUZZY INFERENCE SYSTEM FOR IMPLEMENTATION IN THE FPGA STRUCTURE

BERNARD WYRWOŁ, EDWARD HRYNKIEWICZ

Institute of Electronics
Silesian University of Technology, Akademicka 16, 44-101 Gliwice, Poland
e-mail: {Bernard.Wyrwol,Edward.Hrynkiewicz}@polsl.pl

The paper presents the design and implementation of a digital rule-relational fuzzy logic controller. Classical and decomposed logical structures of fuzzy systems are discussed. The second allows a decrease in the hardware cost of the fuzzy system and in the computing time of the final result (fuzzy or crisp), especially when referring to relational systems. The physical architecture consists of IP modules implemented in an FPGA structure. The modules can be inserted into or removed from the project to get a desirable fuzzy logic controller configuration. The fuzzy inference system implemented in FPGA can operate with a much higher performance than software implementations on standard microcontrollers.

Keywords: fuzzy logic, fuzzy inference algorithm, decomposition, digital fuzzy logic controller, FPGA.

1. Introduction

The general architecture of the Multiple Inputs Single Output (MISO) Fuzzy logic Inference System (FIS) is shown in Fig. 1. It consists of the following components: a fuzzification block, a knowledge base, an inference block and a defuzzification block (Chojcan and Łęski, 2001; Czogała and Pedrycz, 1985; Rutkowska *et al.*, 1997; Kovačić and Bogdan, 2006; Passino and Yurkovich, 1998; Piegat, 2006).

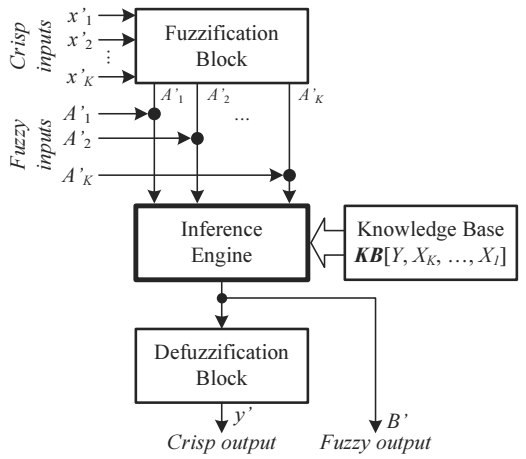


Fig. 1. General architecture of the fuzzy logic inference system.

The knowledge base $\mathbf{KB}[X_K, \ldots, X_1, Y]$ contains a collection of linguistic rules and a definition of linguistic variables. The fuzzy system is characterized by the linguistic description in the form of fuzzy rules,

$$\text{If } X_K \text{ is } A_{Ki_K} \text{ and, } \ldots, \text{and } X_2 \text{ is } A_{2i_2} \text{ and } X_1 \text{ is } A_{1i_1} \text{ then } Y \text{ is } B_{i_K \ldots i_2 i_1}, \quad (1)$$

where $X_K, \ldots, X_2, X_1$ are input variables, Y is an output variable, A_{Ki_K}, …, A_{2i_2}, A_{1i_1}, $B_{i_K \ldots i_2 i_1}$ are linguistic values defined by fuzzy sets (Piegat, 2005) on the corresponding universes of discourse $\mathcal{X}_K, \ldots, \mathcal{X}_2, \mathcal{X}_1$ and $\mathcal{Y}$, respectively ($i_K = 1, \ldots, N_K, \ldots, i_2 = 1, \ldots, N_2$, $i_1 = 1, \ldots, N_1$, where N_k ($k = 1, \ldots, K$) denotes the number of the linguistic values for the k-th input variable).

The general inference process usually proceeds in four (or three for a system with only a fuzzy output) steps (Czogała and Pedrycz, 1985; Rutkowska *et al.*, 1997; Sulaiman *et al.*, 2009):

1. Fuzzification: the membership functions defined on the input variables $x = [x_K, \ldots, x_2, x_1]$ are applied to their actual values $x' = [x'_K, \ldots, x'_2, x'_1]$ to determine the degree of truth for each rule premise (the if-parts of the rules). An operation is superfluous if the input variables are fuzzy ($\mathbf{A}' = [A'_K, \ldots, A'_2, A'_1]$). The most popular method is singleton fuzzification (systems with no fuzzy inputs).

2. Inference: the truth value for the premise of each rule is computed and applied to the conclusion part of each rule (the then-parts of the rules).

3. Aggregation: all of the fuzzy subsets obtained in the previous step are combined together to form a single fuzzy set B for output variable Y (fuzzy output).

4. Defuzzification: converts the fuzzy output set B to a crisp number y (this operation is superfluous if the fuzzy logic inference system has only fuzzy output).

2. Rule and relational fuzzy systems

The output fuzzy set $B_{i_K \dots i_1}$ for rule $R_{i_K \dots i_1}$ can be expressed by means of the formula (Czogała and Pedrycz, 1985; Rutkowska *et al*., 1997)

$$B'_{i_K \dots i_1} = \mathbf{A}' \circ \Re_{i_K \dots i_1}, \tag{2}$$

where the symbol $\circ$ denotes the compositional rule of inference operators (e.g., sup-min, sup-prod), and $\Re_{i_K \dots i_1}$ represents the relation between the premise and antecedent of $R_{i_K \dots i_1}$ rule. The single output fuzzy set B for collection of rules can be computed on the basis of two approximate reasoning methods:

Method 1. The fuzzy sets $B'_{i_K \dots i_1}$ are combined together to get a single fuzzy set by using aggregate operator, denoted as $\dot{\vee}$:

$$B' = \dot{\bigvee_{i_K=1}^{N_K}} \dots \dot{\bigvee_{i_1=1}^{N_1}} B'_{i_K \dots i_1}. \tag{3}$$

Method 2. A global relation $\Re$ for all rules is appointed as

$$\Re = \dot{\bigvee_{i_K=1}^{N_K}} \dots \dot{\bigvee_{i_1=1}^{N_1}} \Re_{i_K \dots i_1}, \tag{4}$$

and then the output fuzzy set is computed according to the formula

$$B' = \mathbf{A}' \circ \Re. \tag{5}$$

In Method 1, Steps 2 and 3 of the algorithm described in Section 1 are always performed when the input values are changed while in Method 2 they are executed when aggregating all rules to get the global relation. Fuzzy systems using the first method are called rule fuzzy systems or FITA (First Inference Then Aggregate), those applying the second one—relational fuzzy systems or FATI (First Aggregate Then Inference) (Czogała and Łęski, 1998).

3. Hardware models of the FITA and FATI systems

In the discussion presented below, it has been assumed that the fuzzy reasoning method is based on Mamdani's composition (conjunctive interpretation of if-then rules). In this case, the relation $\Re$ is of the general form

$$\Re = A \wedge B, \tag{6}$$

where $\wedge$ denotes the MIN operator (Czogała and Łęski, 1998; Rutkowska *et al*., 1997).

The membership function for Method 1 (singleton fuzzification method) can be expressed as

$$\mu_{B'}(y) = \bigvee_{i_K=1}^{N_K} \dots \bigvee_{i_1=1}^{N_1} \left[\tau_{i_K \dots i_1} \wedge \mu_{B_{i_K \dots i_1}}(y) \right], \tag{7}$$

where $\tau_{i_K \dots i_1}$ is a degree of truth for the $i_K \dots i_1$-th rule,

$$\tau_{i_K \dots i_1} = \bigwedge_{j=1}^{K} \mu_{A_{j i_j}}(x'_j). \tag{8}$$

The formula (7) can be computed in the structure presented in Fig. 2 (Hrynkiewicz and Wyrwoł, 2000). It consists of the following components:

- $A_{K i_K}$, $B_{i_K \dots i_1}$: memory modules, store values of membership functions of linguistic values in the if- and then-part of rules, respectively, as a binary matrix,
- $\wedge$: MIN components, obtain truth values of the premises for each rule (second level) or compute a fuzzy subset $B'_{i_K \dots i_1}$ for each rule (third level),
- $\vee$: MAX component, makes an aggregation of fuzzy subsets $B'_{i_K \dots i_1}$ to get the output result B'.

It can be noticed that, using the classical Mamdani inference technique, the FITA inference system, presented in Fig. 2, triggers all rules in every calculation of the output result (Sakthivel *et al*., 2010; Uppalapati and Kaur, 2009; Al-Aubidy, 2010).

The membership function for Method 2 (Czogała and Łęski, 1998; Rutkowska *et al*., 1997; Yager and Filev, 1994) can be expressed as

$$\mu_{B'}(y) = \sup_{x \in \mathcal{X}} \left[\left(\bigwedge_{k=1}^{K} \mu_{A'_k}(x_k) \right) \wedge \mu_{\Re}(x_K, \dots, x_1, y) \right], \tag{9}$$

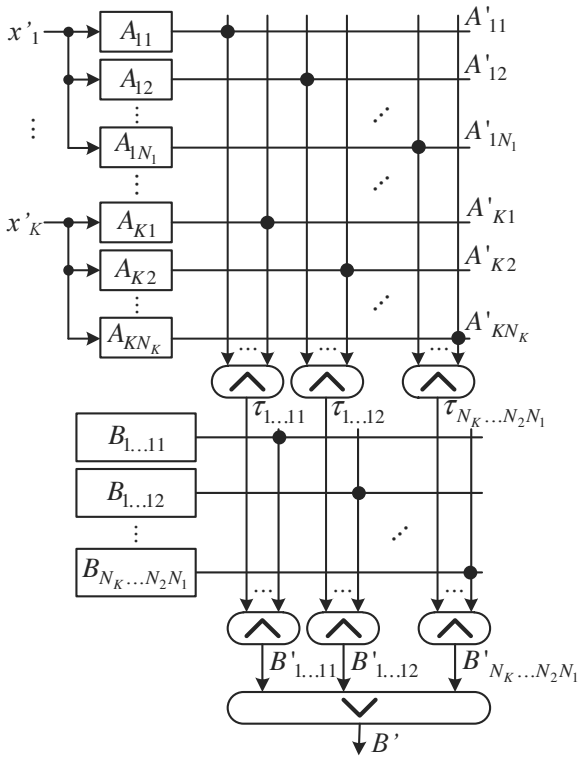


Fig. 2. Hardware model of the FITA fuzzy inference system.

where the membership function of the global fuzzy relation is

$$\mu_{\Re}(x_K,\ldots,x_1,y) = \bigvee_{i_K=1}^{N_K}\cdots\bigvee_{i_1=1}^{N_1}\left[\left(\bigwedge_{k=1}^{K}\mu_{A_{ki_k}}(x_k)\right)\wedge\mu_{B_{i_K\ldots i_1}}(y)\right]. \tag{10}$$

The formula (9) can be evaluated in the structure presented in Fig. 3. The membership function of the global fuzzy relation (4) is computed (before the inference process for input values $\mathbf{x}'$ has been started) and stored in cells of the memory $\Re$ (as a fuzzy look-up table). The fuzzy operations '$\min$' and '$\max$' are performed by decoders and the output buffer of the memory (during the inference process).

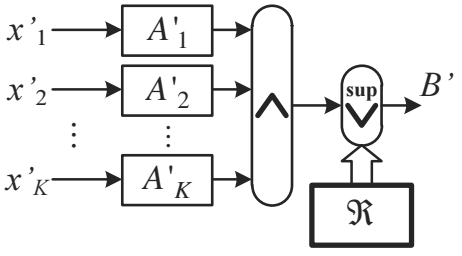


Fig. 3. Hardware model of the FATI fuzzy inference system.

4. Decomposition technique

A decomposition technique based on a projection of the global fuzzy relation has been proposed by Gupta *et al.* (1986). It allows the global relation $\Re$ to be converted into subrelations $\Re_i$ $(i = 1,\ldots,K)$, and thus can be used only in relation type inference systems (FATI)

$$\Re_i = \operatorname*{proj}_{x_i}(\Re), \tag{11}$$

where projection is defined as

$$\operatorname*{proj}_{x_n,\ldots,x_1}(\Re) = \max_{y_n,\ldots,y_1}[\Re(x_n,\ldots,x_1,y_n,\ldots,y_1)]. \tag{12}$$

This technique requires calculating the global fuzzy relation $\Re$ based on information stored in the knowledge base $\mathbf{KB}[X_K,\ldots,X_1,Y]$ of the fuzzy system (Fig. 4). A lot of time is required to compute it and considerable

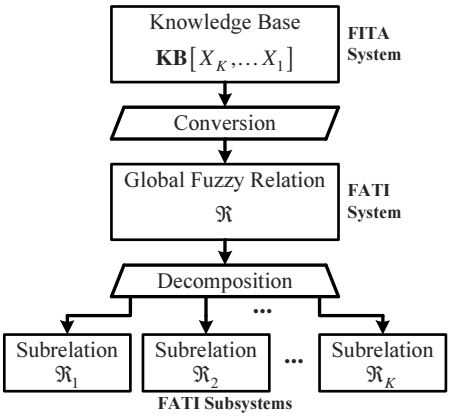


Fig. 4. Flow diagram of creating subrelations (FATI subsystems) based on decomposing the global fuzzy relation.

memory is needed to store it. These disadvantages can be eliminated if decomposition is used for the knowledge base (Walichiewicz, 1984; Martins and Carvalho, 2001; Wyrwoł, 2004a). In this case, Gupta's decomposition method can be extended into FITA systems (Fig. 5).

5. Hierarchical model of the FITA and FATI systems

The general structure of the decomposed fuzzy system is shown in Fig. 6 (for Gupta's primary decomposition method $p = 1$; to avoid the decomposition error (Di Nola *et al.*, 1984; 1985; Lee *et al.*, 1995), using a modified decomposition technique, e.g., based on partitioning the knowledge base $\mathbf{KB}[X_K,\ldots,X_1,Y]$ (Wyrwoł, 2004a; 2008; 2011), the number of subsystems p in general cases can be greater than 1). It consists of p subsystems, each of them made of K SISO (Single Input Single Output)

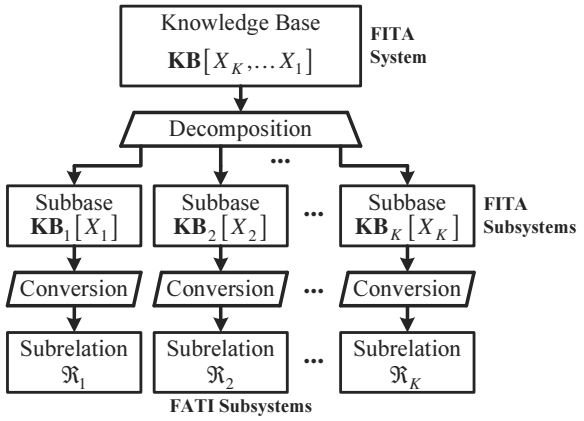


Fig. 5. Flow diagram of creating knowledge bases (FITA subsystems) and then subrelations (FATI subsystems) based on a decomposition of the global knowledge base.

systems (in Fig. 6 marked as FIS_{pk} $(k = 1, \ldots, K)$; p depends on the decomposition method). They can be implemented as rule (FITA) or relational (FATI) fuzzy inference engines.

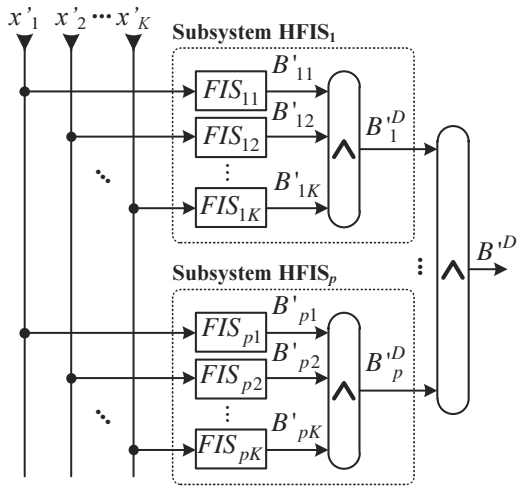


Fig. 6. General architecture of the hierarchical fuzzy inference system.

6. Comparison of the primary and the decomposed model of fuzzy inference systems

The estimated hardware cost of the fuzzy inference system can be expressed as

$$HC \approx HC_{P_{\mathrm{mem}}} + HC_{L_{\mathrm{conn}}} + HC_{L_{\mathrm{com}}}, \qquad (13)$$

where $HC_{P_{\mathrm{mem}}}$, $HC_{L_{\mathrm{conn}}}$ and $HC_{L_{\mathrm{comp}}}$ denote the hardware cost of the memory modules, connections and components used in the system, respectively (Wyrwoł, 2004a). The hardware cost can be calculated for primary (H^{FIS}) and hierarchical (H^{HFIS}) structures of the models described in Sections 3 and 5. To compare the two structures, a hardware cost reduction coefficient has been defined as

$$\upsilon_{HC}[\%] = \frac{H_{FIS} - H_{HFIS}}{H_{FIS}} \cdot 100. \qquad (14)$$

Theoretically, the computed hardware cost reduction coefficient is presented graphically in Fig. 7 for relational systems and Fig. 8 for rule systems. For reasonable parameters n or N ($p = 1$), the decomposition method leads to the lowering of hardware costs (Hung-Ping and Parug, 1996).

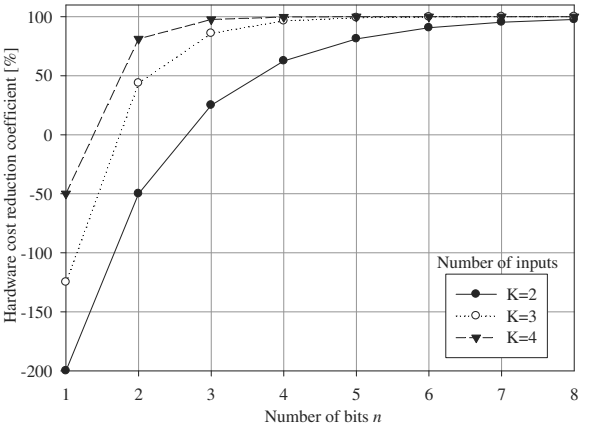


Fig. 7. Hardware cost reduction coefficient vs. the number of bits of input and output values and the number of inputs for relational fuzzy systems (FATI).

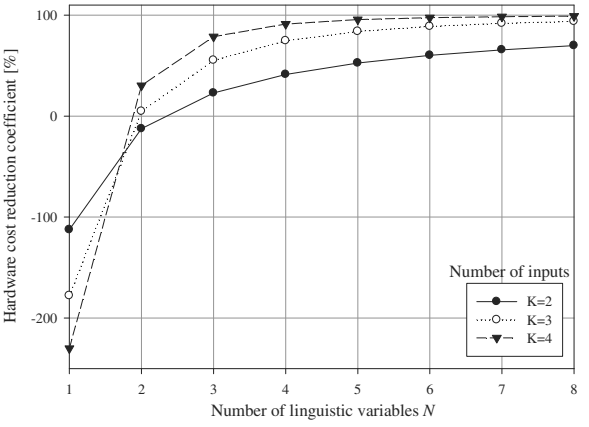


Fig. 8. Hardware cost reduction coefficient vs. the number of linguistic values and the number of inputs for each linguistic variable for rule fuzzy systems (FITA).

The practically created fuzzy inference systems, in the aspect of the hierarchic structure, do not always permit reducing hardware costs, especially if the parameter p is

greater than 1. The hardware cost reduction coefficient for some fuzzy inference systems (used as benchmarks) is presented in Table 1. The knowledge bases of the systems describe respectively fuzzy controllers (denominated as 1, 3, 4) (Baturone *et al.*, 1997; Kim and Cho, 1999; Yager and Filev, 1994), an ENOR gate (denominated as 2) (Lee *et al.*, 1995), a truck park controller (denominated as 5) (Rutkowska *et al.*, 1997; Kim, 2000), a temperature controller of a heated air-stream (denominated as 6) (Ollero and Garcia-Cerezo, 1989), a fuzzy controller for stabilization of an inverted pendulum (denominated as 7) (Yamakawa, 1989), a fan controller (denominated as 8) (Hurdon, 1993) and a fuzzy system for identification of nonlinear systems (denominated as 9) (Rovatti *et al.*, 1995). For the primary decomposition technique ($p = 1$), the hardware cost is lower if the system is built as a hierarchical structure, and it is the highest for FATI systems. The number of subsystems should be increased in some cases ($p > 1$) to avoid the inference error (Lee *et al.*, 1995, Wyrwoł, 2004a; 2008; 2011), and then the hardware cost of the system may increase. This problem is not critical for most FATI systems.

Summarizing, the hierarchical structure of the fuzzy inference analytical model (Section 5) offers major advantages over the flat structure (Section 3):

- lower hardware cost;
- hardware cost (of the FITA system) does not depend strongly on the number of linguistic values of the input variables N_i ($i = 1, \ldots, K$, the formula (1)), e.g., does not depend strongly on the number of rules $N = N_1 \cdot N_2 \cdot \ldots \cdot N_K$;
- system consists of the same simple and compact structure components (SISO subsystems and fuzzy arithmetic logic units).

Table 1. Hardware cost reduction $\upsilon[\%]$ for practically built fuzzy inference systems.

Benchmark	System			
	FATI		FITA	
	$p = 1$	$p > 1$	$p = 1$	$p > 1$
1	98	91	55	16
2	98	94	-4	-38
3	98	86	51	-27
4	98	91	23	-38
5	98	86	47	-21
6	98	91	34	-25
7	98	91	22	-9
8	98	94	23	-6
9	98	91	49	16

7. Hardware structure of the modular fuzzy rule/relational system

From the comparison of the FATI and FITA systems, one can conclude the following:

- hardware cost of the FATI does not depend on the number of rules;
- FATI systems calculate the result of inference in the shortest time;
- FITA systems allow the parameters of the knowledge base to be changed during the inference process (adaptive control systems);
- FITA systems require complex fuzzy logic arithmetic units to be implemented (their hardware cost depends on the format of the membership functions);
- FATI systems require bigger memory to store fuzzy relations (global or subrelations in the case of a hierarchical structure).

Hardware implementation of the rule-relational, modular fuzzy inference system allows high performance (FATI approximate reasoning method), flexibility (altering parameters of the knowledge base, the system architecture, etc.), and additionally, low cost (a hierarchical structure, smaller size of memory required to store fuzzy relations). The general architecture of the digital 8-bit fuzzy inference system FPGA-FIS is shown in Fig. 9 (Wyrwoł, 2004a). It consists of two main components: a memory module and an FPGA chip. The first is connected to the FPGA via an 8-bit bidirectional data bus, 20-bit address bus and 6-bit control bus. The external RAM module is generally used to store the knowledge base of the system (or subsystems) and fuzzy subrelations (as a form of fuzzy look-up tables).

In the FPGA chip module of the fuzzy inference system two interfaces are implemented: Memory Interface and Control/Configuration Interface. The first provides communication between the modules contained in FPGA and external RAM. The second allows an external device (e.g., a microprocessor) to configure the fuzzy system and then to control the inference process.

The modules, implemented in the FPGA chip, perform various tasks: fuzzy operations control the inference process, system configuration, etc. They are provided communication (control signals and data) via an internal bus, but at the same time only one of them is the master (control unit) and has direct access to the RAM buses to control the system behavior.

All of the main designed modules are briefly described below:

MMU (Memory Management Unit): It allows access to RAM memory and supports write and read configuration data of the fuzzy system.

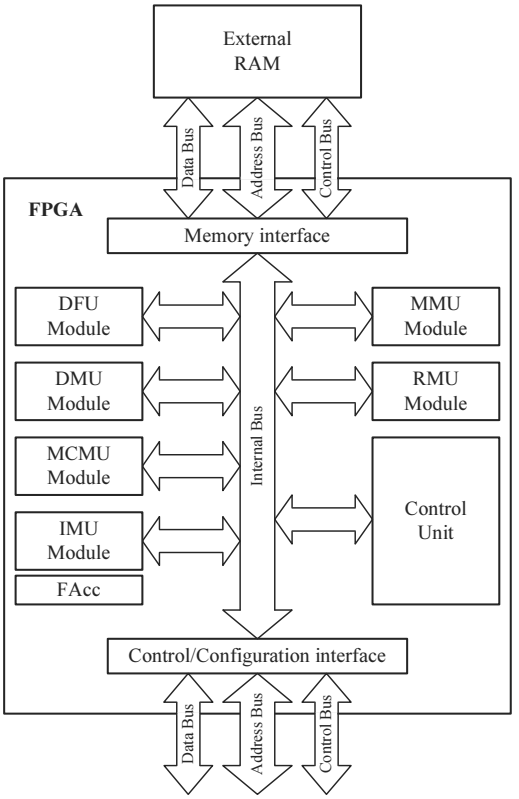


Fig. 9. General architecture of the digital fuzzy logic inference system.

FAcc (Fuzzy Accumulator): It executes a basic operation on fuzzy arguments (membership functions in the binary matrix form).

IMU (Inference Management Unit): It performs and controls an inference process.

DFU (DeFuzzification Unit): It converts a fuzzy inference result into a crisp value using one of the following methods: COG, COA, FOM, LOM and MOM (the module is not required for a system with fuzzy output only).

RMU (Relation Management Unit): It converts information from the knowledge base of the SISO system into the corresponding fuzzy relation.

DMU (Decomposition Management Unit): It allows the decomposing of the knowledge base of the primary system into a knowledge base of the SISO subsystems.

MCMU (Membership Conversion Management Unit): It converts a parametric membership function into a look-up table.

Listing 1. Example description of Memory Management Unit in Verilog HDL.

```
// ports list
module MMU(// Ready signal (output)
             Ready,
           // Strobe signal (input)
             Stb,
           // Data direction (Read/notWrite) (input)
             RW,
           // Module initialization (input)
             Init,
           // Memory write and reas signals (outputs)
             MemWE, MemRE,
           // Address buses (outputs)
             AddrX, AddrY, AddrS,
           // Number of subsystems (p) (outputs)
             NSubS,
           // RAM module selection bus (inputs)
             SelM);

// parameters declaration
parameter DataMBF = 8, DataIO = 8;

// input ports declaration
input Stb;
input RW;
input Init;
input [3:0]SelM;

// output ports declaration
output MemWE;
output MemRE;
output [DataMBF-1:0]AddrX; reg [DataMBF-1:0]AddrX;
output [DataMBF-1:0]AddrY; reg [DataMBF-1:0]AddrY;
output [3:0]AddrS;         reg [3:0]AddrS;
output Ready;              reg Ready;
output [3:0]NSubS;         reg [3:0]NSubS;

// continuous assignment
assign MemRE = ~RW;
assign MemWE = RW | Stb;

// structural module
always @(posedge Stb or posedge Init) begin
  if (Init==1)
    // MMU initialization
    begin
      // address bus
      AddrX = 8'b00000000;
      AddrY = 8'b00000000;
      AddrS = SelM;
      Ready = 1;
    end
  else
    // MMU run
    begin
      // increment address
      {AddrS[3:0], AddrX[7:0], AddrY[7:0]} =
        {AddrS[3:0], AddrX[7:0], AddrY[7:0]} + 1;
      // signal Ready
      Ready = ~({AddrX[7:0], AddrY[7:0]}==0);
      // number of used RAM modules
      if ((Ready==0) && (RW==0)) NSubS = AddrS;
    end
end

endmodule
```

The modules can be implemented in an FPGA chip to create a desirable rule (FITA), relational (FATI) or rule-relational (FITA-FATI) fuzzy system (Table 2: '$*$' denotes that the module is not required for a system with fuzzy output only, '$\bullet$' means that the module is always required in the system, '$\circ$' means that the module can be used in the system, but if not implemented, the appropriate task has to be executed by an external device, e.g., in a microprocessor system). If any optional component is

not implemented in the fuzzy system, the appropriate task (for example, calculating a fuzzy relation) should be executed by an external device (and the final results, for example, a fuzzy relation, are then stored in the RAM of the system). The library of modules has been described in Verilog HDL (Accellera, 2002; Xilinx, 2009; Bhasker, 1998; Palnitkar, 1996; Minns and Elliott, 2008). This allows implementing it in any FPGA chip easily (the modules can be used in design entry phase of the system). An example of the description of one of the modules, Memory Management Unit, is depicted in Listing 1.

8. Example implementation of the fuzzy relational system

As an example, possible implementation of the fuzzy relational system (FATI) with 8-bit resolution is described. For clarity the system, has been divided into two separate parts. They are illustrated in Figs. 10 and 12.

The first shows part of a fuzzy system which is active during the configuration process, the second—during the inference process. The master module in the configuration mode is Memory Management Unit (Listing 1). It provides all the necessary signals and an address to write (or read) information from an external device to the RAM modules. The external device, e.g., a PC with a dedicated program, prepares fuzzy subrelations, according to Eqn. (11) and Fig. 5, and it sends to the memory the modules of the FPGA-FIS system (the fuzzy subrelations can be also created in the system using additional modules gathered in Table 2). Configuration data (subrelations) are sent to the inference system via an RS232 interface. Therefore, an additional microcontroller has been used. It converts serial data into parallel data, accepted by the Control/Configuration Interface.

The data to be sent to the system are organized in blocks of 64 kB. Each data block represents a subrelation for subsystem FIS_{pk} ($k = 1, \ldots, K$; p depends on the decomposition method, in some cases p is equal to 1), as depicted in Fig. 6. For a SISO system, the subrelation

Table 2. Modules required for realization of the specific fuzzy inference system.

Module	System		
	FITA	FATI	MIX
MMU	•	•	•
FAcc	•	•	•
IMU	•	•	•
DFU	∗	∗	∗
RMU		○	○
DMU		○	○
MCMU	○	○	○

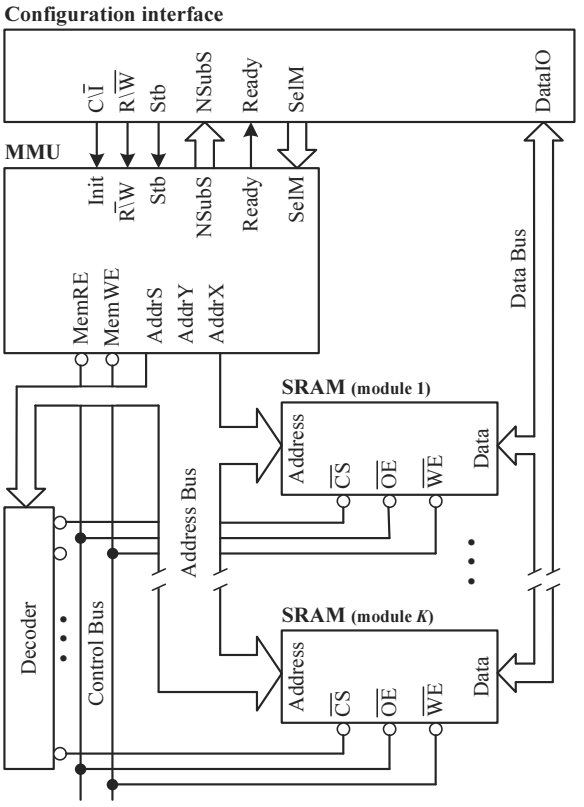


Fig. 10. One of the possible hardware configurations of the fuzzy inference system (initialization mode).

requires

$$C_{\mathrm{RAM}}[\mathrm{bits}] = mbf\_res \cdot 2^{(x\_res+y\_res)} \qquad (15)$$

of memory, where mbf_res, x_res and y_res are membership, input and output data resolutions, respectively.

The master module in the inference mode is the Inference Management Unit (Fig. 12). It provides an address to the RAM modules (the memory is in read mode) and necessary signals to Fuzzy Accumulator FAcc and Defuzzification Unit DFU.

The main function of the FAcc module is to find the fuzzy output set B' by computing the fuzzy AND operation of the fuzzy sets $B'_1, \ldots, B'_K$ (results of the composition actual fuzzified input values $x_1, \ldots, x'_K$ and fuzzy relations stored in RAM, as expressed by Eqn. (9)). As an example, the description of a simplified version of the two-input Fuzzy Accumulator FAcc2 is depicted in Listing 2. It has been assumed that the membership functions of the fuzzy sets (as well as functions of fuzzy relations) have the form of a look-up table as presented in Fig. 13 (Patyra *et al.*, 1996).

Defuzzification Unit transfers the fuzzy inference result B' to the external device via Control/Configuration Interface or converts it first into a crisp value. The

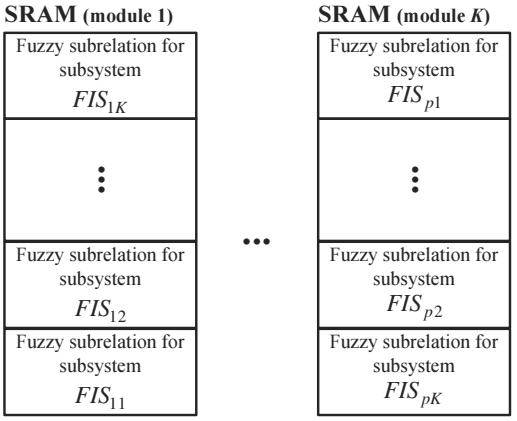


Fig. 11. Memory map of the relational fuzzy inference system.

inference and defuzzification tasks can be executed simultaneously (Wyrwoł, 2004b).

Listing 2. Simplified description of the FAcc unit in Verilog HDL (for a two-input fuzzy inference system).

```
// ports list
module FAcc2(// input data D0 (input)
                DataD0,
             // input data D1 (input)
                DataD1,
             // output data DO (output)
                DataO,
             // clock signal (input)
                Clk,
             // reset signal (input)
                Reset);

// parameters declaration
parameter DataMBF = 8;

// input ports declaration
input Clk;
input Reset;
input [DataMBF-1:0] DataD0;
input [DataMBF-1:0] DataD1;

// output ports declaration
output [DataMBF-1:0] DataO; reg [DataMBF-1:0] DataO;

// temporary registers
reg [DataMBF-1:0] DataMin;

// structural module
always @(posedge Clk or posedge Reset) begin
  if (Clk)
    begin
      // AND operation on inputs data
      DataMin = (DataD0>DataD1) ? DataD1 : DataD0;
      if (DataMin>DataO) DataO = DataMin;
    end
  // reset output
  else DataO=0;
end

endmodule
```

9. Conclusion

The fuzzy inference system presented in the paper has been tested on a prototype board. It consists of an FPGA Xilinx Spartan II XC2S200 chip (Xilinx, 2008), an Atmel

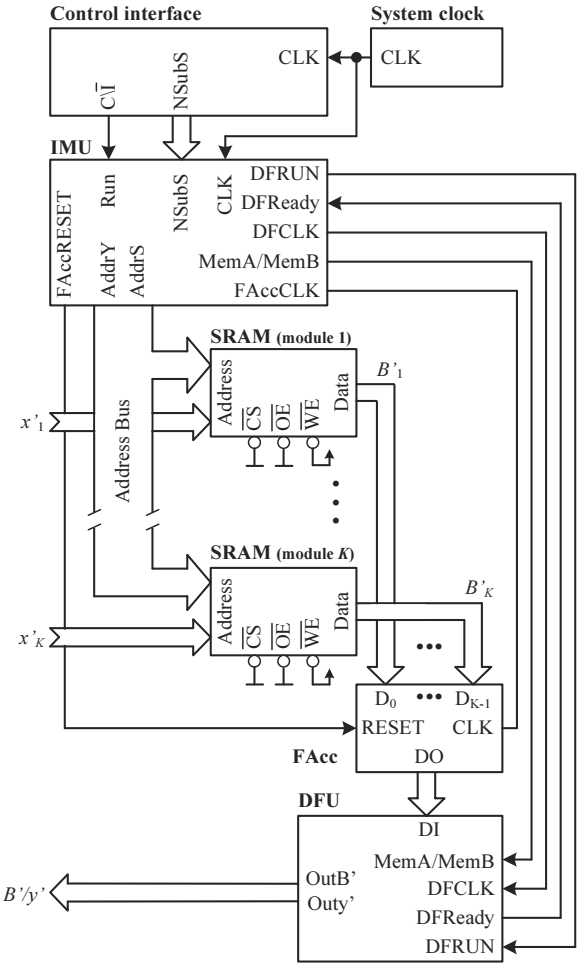


Fig. 12. One of the possible hardware configurations of the fuzzy inference system (inference mode).

AVR ATMega family ATMega32 microcontroller (Atmel, 2007) with an RS232 interface and 1 MB of external SRAM on the board (2 modules K6T4008C1B, (Samsung Electronics, 1998)). The microcontroller is connected to Control/Configuration Interface of the FPGA. It operates only as an RS232 monitor. It receives commands or data from a host computer (FPGA-FIS software, not described in the paper, allows the configuration data of the fuzzy system to be prepared, controlled and tested) and sends it to the FPGA. Additionally, the development system has an on-board DLC5 ISP programmer (Zbysiński and Pasierbiński, 1992), which allows loading the bitstream of a design as generated by the Xilinx development software WebPack ISE (ver. 8) into the internal configuration memory of the FPGA.

All of the modules presented in Table 2 have been implemented and tested using the XC2S200 prototype board. The hardware resources of the FPGA chip required for implementation of the modules are presented in Table 3.

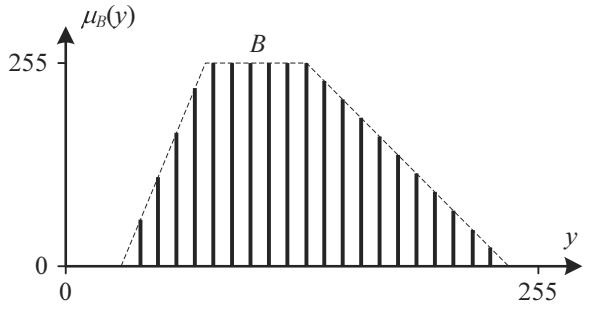


Fig. 13. Membership function format of the example fuzzy set B (8-bit resolution).

Table 3. Hardware resources required of implementation of fuzzy system modules.

Module	Number of			Total equivalent gate count
	Slices	F-Fs	LUTs	
MMU	38	27	43	567
FAcc	25	8	33	310
IMU	26	20	40	427
DFU[1]	240	74	457	4205
RMU	85	56	137	1372
DMU	112	65	207	1831
MCMU[2]	156	88	259	2597

[1] The module executes the COG, COA, FOM, LOM or MOM defuzzification method.
[2] The module converts the parametric membership function Gamma, L, T type, into a look-up table.

The hardware fuzzy inference system can be characterized by hardware cost and performance. The first parameter was discussed in Section 6. The system based on an FPGA chip has 1 MB of external SRAM. It is sufficient to store the subrelations of the system depicted in Fig. 6, knowledge bases of the primary system and decomposed SISO subsystems. As an example, the fuzzy relational system has two-input, single-output and 8-bit data resolution. The classical implementation of the system requires up to 16 MB of RAM. Considering the system to be designed as a hierarchical architecture, the memory amount has been reduced to 2×64 kB=128 kB (according to Eqn. (15)).

Performance can be characterized by the input to output time t_{IOdt} (Patyra *et al.*, 1996; Chmiel and Hrynkiewicz, 2008). This is defined as the time from the moment of providing the input variables to the system until computing the output result (crisp or fuzzy) at the output. The performance of some digital fuzzy inference systems (PLC Simatic S7 CPU416 and CPU314 (Siemens AG, 1996), FPGA XC4006 (Hollstein *et al.*, 1996), ASIC FC110 (Togai InfraLogic, Inc., 1991; Hollstein *et al.*, 1996), DDS Fuzzy Logic (Patyra *et al.*, 1996)) is presented in Table 4.

It can be noted that in SimaticS7 (CPU416 and CPU314) a fuzzy inference system is implemented in PLC hardware in a program way (as an FB30, FC30 or FC31 modules). Hence, the data are processed serially and the performance of the system is the lowest. The other systems are implemented in hardware and the performance is higher. The DDS Fuzzy Logic System is characterized by the highest performance, but its hardware cost depends strongly on input and output variables' resolution. Thus the variables are 4-bit in length and are not enough for most practically realized applications. The other systems, gathered in Table 4, operate on 8-bit length data and the fuzzy engine is implemented as a rule system (FITA).

It should also be noted that FPGA-FIS (for relational and rule-relational version of the system) performance is constant, does not depend on configuration parameters of the fuzzy system (e.g., number of if-then rules) and it is limited only by the external memory access time (55 ns, (Samsung Electronics, 1998)). Theoretically, the input to output delay can be decreased to 15 ns by increasing the frequency of the system clock (maximum frequency for the FIS project implemented in an FPGA chip is equal to 36 MHz).

In conclusion, the presented digital, modular, hierarchical fuzzy inference system offers these major advantages:

- Modular architecture allows an appropriate rule (FITA), relational (FATI) or rule-relational (FITA-FATI) fuzzy system to be designed.
- Easy configuration of the system (the design entry phase by coding the system in an HDL or by a schematic representation) using fuzzy components from an IP library (Table 2).
- Architecture of the system can easily be changed through downloading the project data stream into the internal configuration SRAM of the FPGA (the

Table 4. Performance of various fuzzy inference systems.

FIS	$t_{\mathrm{IOdt}}[\mu s]$	Remarks
Simatic S7 CPU416	700– –3000	for FB30, FC30, FC31 fuzzy modules
Simatic S7 CPU314	3000– –13000	for FB30, FC30, FC31 fuzzy modules
FPGA XC4006	42	4-bit system
ASIC FC110	32	8-bit fuzzy processor, for a 20 MHz system clock
DDS Fuzzy Logic	0.05	4-bit system, parallel architecture implemented in ASIC chip
FPGA– –FIS	21	for a 24 MHz system clock

reconfiguration property of an FPGA chip is not allowed for the current version of the proposed fuzzy inference system),

- High performance (Table 4).
- Performance does not depend on the kind of output result (crisp or fuzzy) and the number of rules in the knowledge base (only for a configuration relational (FATI) or rule-relational system (FITA-FATI)).
- Parameters of the knowledge base can easily be changed, also for relational (FATI) or rule-relational (FITA-FATI) system configuration.
- Low cost, size of the memory to store fuzzy relations is the smallest.

For future research, the system will be implemented as a PSOC (Programmable System On Chip) device, the library will be expanded with new modules and the reconfigurable property of the FPGA chip will be used to dynamically change the configuration of the system in the configuration and inference modes.

References

Accellera (2002). *System Verilog 3.1*, `www.eda.org/sv/SystemVerilog_3.1a.pdf`.

Al-Aubidy, K.M. (2010). FPGA-based fuzzy inference system for real-time embedded applications, *International Journal of Real-Time Systems* **1**(1): 9–15.

Atmel (2007). *8-bit AVR Microcontroller with 32K Bytes In–System Programmable Flash*, 2503–avr–08/07 Edn., ATMEL, `www.atmel.com/Images/doc2503.pdf`.

Baturone, I., Sánchez-Solano, S., Barriga, A. and Huertas, J.L. (1997). Implementation of CMOS fuzzy controllers as mixed-signal integrated circuits, *IEEE Transactions on Fuzzy Systems* **5**(1): 1–19.

Bhasker, J. (1998). *Verilog HDL Synthesis a Practical Primer*, Star Galaxy Publishing, Allentown, PA.

Chmiel, M. and Hrynkiewicz, E. (2008). Fast operating bit-byte PLC, *17th World Congress of the International Federation of Automatic Control, Seoul, Korea*, pp. 14810–14815.

Chojcan, J. and Łęski, J. (2001). *Fuzzy Sets and Their Applications*, Silesian University of Technology Press, Gliwice, (in Polish).

Czogała, E. and Łęski, J. (1998). An equivalence of inference results under defuzzification using both conjunction and logical implication interpretation of fuzzy if-then rules, *6th European Congress on Intelligent Techniques and Soft Computing, 1998, Aachen, Germany*, pp. 83–92.

Czogała, E. and Pedrycz, W. (1985). *Elements and Methods of Fuzzy Set Theory*, Polish Scientific Publishers, PWN, Warsaw.

Di Nola, A., Pedrycz, W. and Sessa, S. (1984). Decomposition problem of fuzzy relations, *International Journal of General Systems* **10**(2–3): 123–133.

Di Nola, A., Pedrycz, W. and Sessa, S. (1985). When is a fuzzy relation decomposable into two fuzzy set, *Fuzzy Sets and Systems* **16**(1): 87–90.

Gupta, M.M., Kiszka, J.B. and Trojan, G.M. (1986). Multivariable structure of fuzzy control systems, *IEEE Transactions on Systems, Man and Cybernetics* **16**(5): 638–656.

Hollstein, T., Halgamuge, S.K. and Glesner, M. (1996). Computer-aided design of fuzzy systems based on generic VHDL specifications, *IEEE Transactions on Fuzzy Systems* **4**(4): 403–417.

Hrynkiewicz, E. and Wyrwoł, B. (2000). Hardware implementation of the FITA fuzzy logic inference systems, *Design and Diagnostics of Electronic Circuits and Systems, DDECS, 2000, Smolenice, Slovakia*, pp. 169–173.

Hung-Ping, C. and Parug, T.-M. (1996). A new approach of multi-stage fuzzy logic inference, *Fuzzy Sets and Systems* **78**(1): 51–72.

Hurdon, H.D. (1993). The fuzzy logic expert fan controller, `www.ecst.csuchico.edu/~juliano/Fuzzy/fuzzyfan`.

Kim, D. (2000). An implementation of fuzzy logic controller on the reconfigurable FPGA system, *IEEE Transactions on Industrial Electronics* **47**(3): 703–715.

Kim, D. and Cho, I.-H. (1999). An accurate and cost-effective COG defuzzifier without the multiplier and the divider, *Fuzzy Sets and Systems* **104**(2): 229–244.

Kovačić, Z. and Bogdan, S. (2006). *Fuzzy Controller Design Theory and Applications*, Taylor & Francis Group, LLC, New York, NY.

Lee, P.G., Kyun, K.L. and Jeon, G.J. (1995). An index of applicability for the decomposition method of multivariable fuzzy systems, *IEEE Transactions on Fuzzy Systems* **3**(3): 364–369.

Martins, A.P. and Carvalho, A.S. (2001). Fuzzy controllers with reduced rulebases and real-time capability for power systems supervision, *Electric Power Components and Systems* **29**(12): 1145–1159.

Minns, P. and Elliott, I. (2008). *FSM-based Digital Design Using Verilog HDL*, John Wiley & Sons, Ltd, New York, NY.

Ollero, A. and Garcia-Cerezo, A.J. (1989). Direct digital control, auto-tuning and supervision using fuzzy logic, *Fuzzy Sets and Systems* **30**(2): 135–153.

Palnitkar, S. (1996). *Verilog HDL: A Guide to Digital Design and Synthesis*, SunSoft Press, Upper Saddle River, NJ.

Passino, K.M. and Yurkovich, S. (1998). *Fuzzy Control*, Addison-Wesley Longman, Inc., Menlo Park, CA.

Patyra, M.J., Gartner, J.L. and Koster, K. (1996). Digital fuzzy logic controller: Design and implementation, *IEEE Transactions on Fuzzy Systems* **4**(4): 439–459.

Piegat, A. (2005). A new definition of the fuzzy set, *International Journal of Applied Mathematics and Computer Science* **15**(1): 125–140.

Piegat, A. (2006). What is not clear in fuzzy control systems?, *International Journal of Applied Mathematics and Computer Science* **16**(1): 37–49.

Rovatti, R., Guerrieri, R. and Baccarani, G. (1995). An enhanced two-level boolean synthesis methodology for fuzzy rules minimization, *IEEE Transactions on Fuzzy Systems* **3**(3): 288–299.

Rutkowska, D., Piliński, M. and Rutkowski, L. (1997). *Neural Networks, Genetic Algorithms and Fuzzy Systems*, Polish Scientific Publishers, PWN, Warsaw.

Sakthivel, G., Anandhi, T.S. and Natarajan, S.P. (2010). Design of optimized fuzzy logic controller for area minimization and its FPGA implementation, *International Journal of Computer Science and Network Security* **10**(8): 187–192.

Samsung Electronics (1998). *K6T4008C1B Family*, 3rd Edn., `www.100y.com.tw/pdf_file/K6T4008C1B-GP70.pdf`.

Siemens AG (1996). *Simatic S7—Fuzzy Control, User Manual*.

Sulaiman, N., Obaid, Z.A., Marhaban, M.H. and Hamidon, M.N. (2009). FPGA-based fuzzy logic—design and applications: A review, *International Journal of Engineering and Technology* **1**(5): 491–503.

Togai InfraLogic, Inc. (1991). *FC 110 Digital Fuzzy Processor DFP^{TM}*, Irvine, CA.

Uppalapati, S. and Kaur, D. (2009). Design and implementation of a Mamdani fuzzy inference system on an FPGA, *28th North American Fuzzy Information Processing Society Annual Conference, NAFIPS2009, Cincinnati, OH, USA*, pp. 1–6.

Walichiewicz, Ł. (1984). Decomposition of linguistic rules in the design of a multi-dimensional fuzzy control algorithm, *Cybernetics and Systems Research, Vienna, Austria*, Vol. 2, pp. 557–561.

Wyrwoł, B. (2004a). *Hardware Implementation of the Fuzzy Inference System Using Programmable Logic Devices*, Ph.D. thesis, Silesian University of Technology, Gliwice.

Wyrwoł, B. (2004b). Modular fuzzy inference system: Compact defuzzyfication module, *7th Conference on Reprogrammable Digital Circuits, RUC'2004, Szczecin, Poland*, pp. 217–224.

Wyrwoł, B. (2008). Linguistic decomposition technique based on partitioning the knowledge base of the fuzzy inference system, *Bulletin of the Polish Academy of Sciences: Technical Sciences* **56**(1): 71–76.

Wyrwoł, B. (2011). Using graph greedy coloring algorithms in the hardware implementation of the HFIS fuzzy inference system, *Electrical Review* **87**(10): 64–67.

Xilinx (2009). *Synthesis and Simulation Design Guide*, `www.xilinx.com/support/documentation/sw_manuals/xilinx11/sim.pdf`.

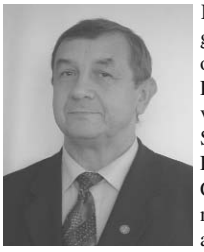
Xilinx (2008). *DS-001 Spartan II—2,5V FPGA Family*, 2.8 Edn. `www.xilinx.com/support/documentation/data_sheets/ds001.pdf`.

Yager, R.R. and Filev, D.P. (1994). *Essential of Fuzzy Modelling and Control*, John Wiley and Sons, New York, NY.

Yamakawa, T. (1989). Stabilization of an inverted pendulum by a high-speed fuzzy logic controller hardware system, *Fuzzy Sets and Systems* **32**(2): 161–180.

Zbysiński, P. and Pasierbiński, J. (1992). *Programmable Devices—First Steps*, BTC Publishing House, Warsaw.

Bernard Wyrwoł received the M.Sc. degree in electronics from the Silesian University of Technology in 1995 and the Ph.D. degree from the same university in 2004, with research on the architecture optimization technique of fuzzy inference systems. He is an assistant professor at the Institute of Electronics, Silesian University of Technology. His research interests include hardware and software implementation of fuzzy inference systems, computer control engineering, programmable logic devices, hardware description languages and multiprocessor systems.



Edward Hrynkiewicz received the M.Sc. degree in electronics from the Silesian University of Technology, Gliwice, Poland in 1971, and the Ph.D. and D.Sc. degrees in electronics respectively in 1978 in 1992 from the same university. Since 1972 he has held various positions at the Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology. Now he occupies a professorial position and is the director of the Institute of Electronics. His primary research interests include logic synthesis, decomposition of logic circuits, programmable devices and systems and utilization of rectangular functions in logic circuits design. He is a member of the Components and Technologies for Control Technical Committee of the IFAC, the IEEE and the Polish Chapter of the IEEE Computer Society, as well as the Electronics and Telecommunication Committee of the Polish Academy of Sciences.

Received: 10 May 2012
Revised: 17 August 2012