A DESIGN METHODOLOGY FOR APPLICATION SPECIFIC FUZZY INTEGRATED CIRCUITS

A. Barriga, R. Senhadji, C. J. Jiménez, I. Baturone, S. Sánchez-Solano

Instituto de Microelectrónica de Sevilla - Centro Nacional de Microelectrónica
Avda. Reina Mercedes s/n, (Edif. CICA)
E-41012, Sevilla, Spain

IEEE International Conference on Electronics, Circuits and Systems (ICECS'98), Vol. 1, pp. 431-434, Lisboa, September 7 - 10, 1998.

A Design Methodology For Application Specific Fuzzy Integrated Circuits

A. Barriga, R. Senhadji, C. J. Jiménez, I. Baturone, S. Sánchez-Solano

Instituto de Microelectrónica de Sevilla - Centro Nacional de Microelectrónica
Avda. Reina Mercedes, s/n Edif. CICA. E-41012 Sevilla, Spain
e-mail: barriga@imse.cnm.es

Abstract

The main objective of this contribution is to present a design methodology for application specific fuzzy integrated circuits. This methodology is based on an specific architecture and a user-friendly design environment which enables the specification, verification and synthesis of fuzzy systems taking into account conceptual as well as microelectronics considerations.

1. Introduction

Nowadays, several circuits known as fuzzy coprocessors are available in the microelectronics market-place. They are general-purpose devices that work together with standard processors to speed up some of the typical operations of fuzzy-logic-based inference systems [1]. The problem is that this kind of circuits is not efficient enough in terms of silicon area, power consumption and inference speed when considering its application to industrial sectors related to telecommunication and automotive or consumer products. A more advantageous solution, as happens in many other microelectronics areas, is to use dedicated hardware adapted to the particular problem [2]. Viability of this approach is greatly increased by using design methodologies, circuit techniques and CAD tools that ease system realization, thus reducing time-to-market.

A design methodology for application specific fuzzy integrated circuits is outlined in this paper. The methodology is based on a set of CAD tools which ease the different stages of specification, verification and synthesis of fuzzy systems [3]. The proposed approach has been validated by applying it to the design of a fuzzy controller for the classical ball and fan problem.

2. Design Methodology

The main steps to design a fuzzy controller following the proposed methodology are illustrated in Fig. 1.

The starting point is the specification of the controller functionality. For this purpose, the design environment provides a flexible high-level specification language, called XFL, that permits a unified fuzzy system description to be held through all the design process. An XFL specification defines the inference mechanism, the defuzzification method and the knowledge base selected (the rule base as well as the antecedents' and consequents' membership functions). An XFL specification consists of type and module definitions that describe the membership functions and the rules, respectively. XFL supports a set of predefined membership function classes as well as a method for defining generic piecewise linear functions. It also supports the use of hierarchical rulebases by employing serial and parallel composition operators [4].

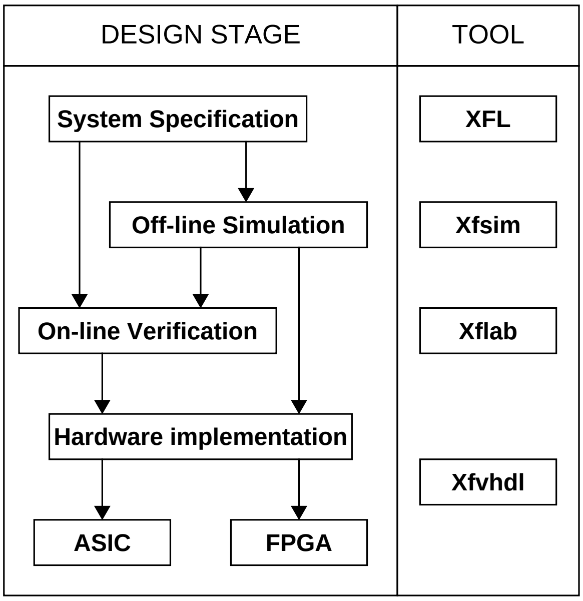


Fig. 1. Design methodology and related tools.

Having made these selections, the user has two ways of verifying the functionality of the whole system (the fuzzy controller and the plant). One of them is to simulate the system behavior with the fuzzy system simulator *xfsim* which is included within the design environment. In this case a software model for the plant is used together with the XFL description. Depending on the problem this first-approximation model can not be enough to describe the plant behavior. A more realistic approach is to include the plant itself within the simulation loop ("hardware in the loop" strategy). For this purpose, the design environment provides the *xflab* utility which basically consists of a data acquisition card and drivers to read and write data. Using this tool the plant model is replaced by a function that performs the following: (a) it monitors the real plant (reading the input ports of the data acquisition card), (b) it sends the inputs excitation to the XFL controller model, and, finally, (c) it writes the control action to the output port of the acquisition card.

Once the controller specifications have been validated, the implementation phase can start. To accomplish this task an adequate controller architecture should have been selected in the XFL specification. The architecture we are currently using enables low-cost and high-inference-speed silicon implementations of fuzzy logic controllers [5]. This is achieved by applying some simplification concepts like the following: (a) the use of an active rule driven inference mechanism, which only explores the (potentially) active rules instead of the whole rule base, (b) the limitations on the membership functions' overlapping degrees so that the membership function circuits (MFC's) can be simplified by reducing the required memories, and (c) the use of crisp consequents, suitable for simplified defuzzification methods, thus exploring the universe of the rules instead of the universe of discourse (Fig. 2).

An automatic synthesis tool, called *xfvhdl*, can be used to translate the XFL specification into a VHDL description capable of being implemented as a microelectronics circuit [6]. In addition to the selection of the actual size of the controller buses, the designer can choose among several implementation options when using *xfvhdl*: (a) vectorial or arithmetic approaches to implement the MFC's, (b) RAM's or ROM's (that increase the programmability) or combinational blocks (that reduce hardware resources) to store the knowledge base, and (c) four different simplified defuzzification methods are currently supported to obtain the crisp controller output.

The output of *xfvhdl* is a set of VHDL files containing the structural controller description, some constant packages, and a test bench file which helps the simulation of the circuit. The description of some of the controller components are directly generated by *xfvhdl,* while others are included into a VHDL cell library. The code used in the description of the cell library is compatible with the restricted VHDL implementations of Synopsys and Mentor Graphics tools.

The VHDL description generated by *xfvhdl* can be synthesized by any conventional high-level synthesis tool. Two generic implementation techniques can be followed. The first one is to use a field programmable gate array (FPGA) [7]. This approach is useful for prototyping purpose because of its inherent capability to change the system functionality by programming. In this case, *xfvhdl* also produces a script file to control the synthesis process with Synopsys and Xilinx to implement the circuit as an FPGA.

The second approach is to design an application specific integrated circuit (ASIC) using a semi-custom IC design environment and the technology library provided by a silicon foundry [8].

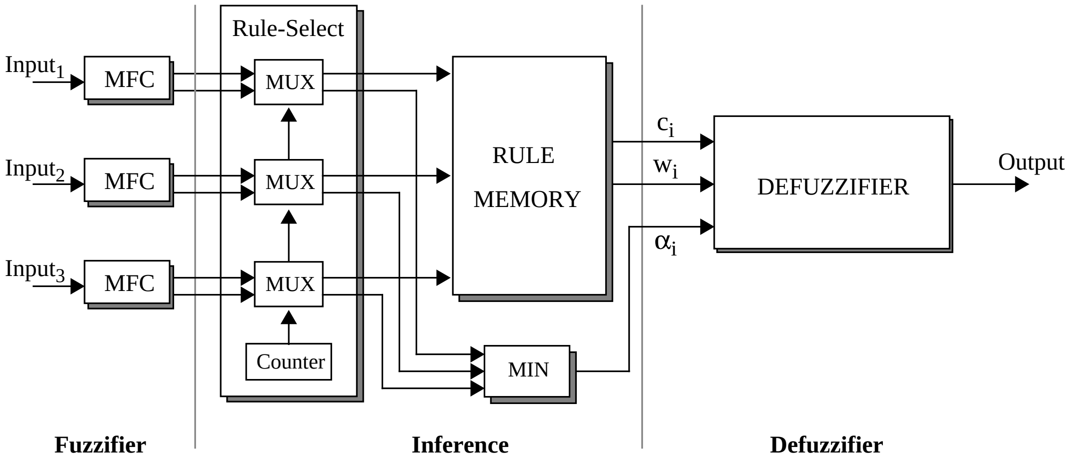


Fig. 2. Active-rule driven architecture for fuzzy controllers.

3. Application Case

In order to illustrate the design flow we describe in this section the design of a fuzzy controller aimed at controlling the height of a falling ball by using a fan. The set up employed in our experiments is shown in Fig 3. It contains: (a) the plant under control, (b) a printed circuit board which includes the fuzzy controller and the circuitry for signal and power conditioning, and (c) a personal computer with a data acquisition card, used as the hardware platform to execute the design and verification tools.

The plant consists of a ball inside a crystal tube that is open at the top and the bottom. The trend of the ball to fall by the gravity action is counteracted by the action of a fan placed at the bottom of the tube. The objective is to hold the ball at a selected height. The position of the ball is detected by a collection of photosensor diodes arranged along the tube. The objective position is fixed in the printed circuit board by activating the adequate switches that codify the position. The output of the fuzzy controller is a digital width modulated pulse signal that governs the fan speed.

Following the methodology described in the previous section, the synthesis process begins with the XFL specification of the fuzzy controller. In this case, the controller acts as a fuzzy PID controller with the total, differential and integral errors as inputs, and the fan speed control signal as output. The fuzzy sets used for antecedents and a control surface plot are shown in Fig. 4.

The next step is to analyze the behavior of the plant inside the closed control loop. To beginning with, the plant is modeled by a program written in C. The tasks of interacting with the model and the presentation of the results are carried out with the help of a graphical user interface, based on the Tcl/Tk programming tools (Fig. 5).

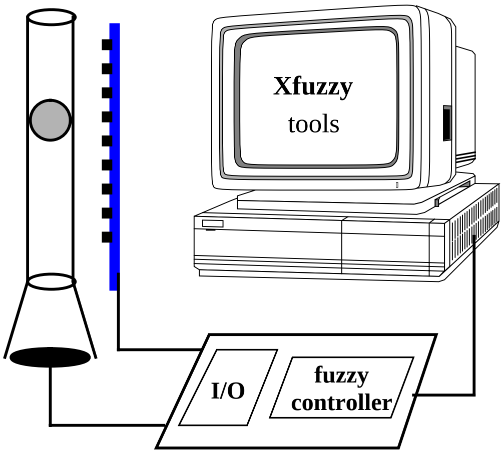


Fig. 3. Experimental set for the application case.

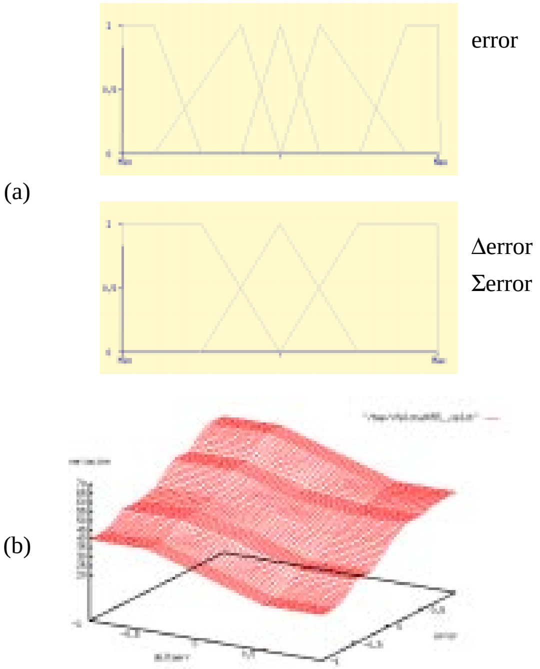
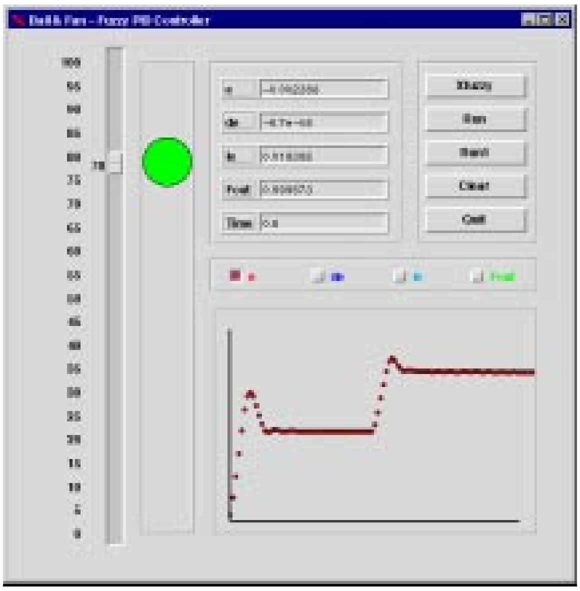


Fig. 4. Controller specification: a) fuzzy sets, b) control surface.

The plant model employed in the simulation stage is usually a first-order model that does not consider effects like friction or turbulence that are very difficult to describe. Thence, the simulation results can be very different from the real ones if these second-order effects are not negligible. To solve this problem the proposed design environment offers an utility to include the real plant into the simulation loop, thus providing an on-line verification of the control.



e -0.062356
de -6.7e-03
ie 0.518392
Fout 0.808573
Time 0.0

Fig. 5. Graphical interface used in the simulation stage.

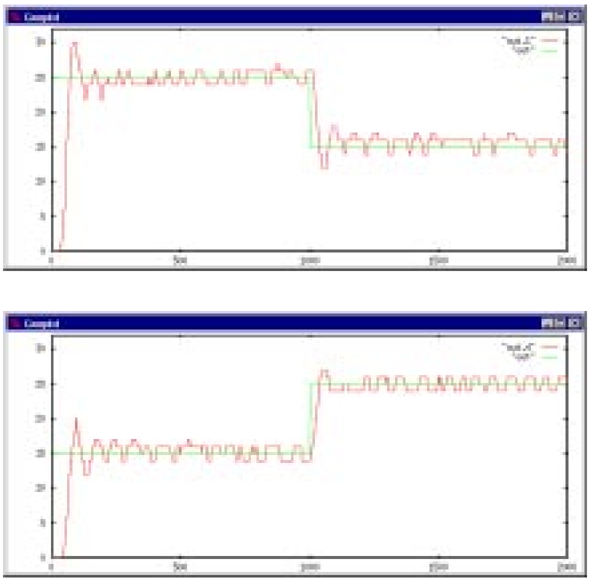


Fig. 6. Controller results in the on-line verification stage.

Fig. 6 illustrates some results of this on-line verification. The graphics represent the temporal evolution of the ball position including several perturbations from an equilibrium point. The position can be seen as quantized into levels that correspond to the distance between two subsequent photosensors.

The last stage of the design process is to implement the verified controller in hardware using the synthesis tools of the design environment. For the application here described the fuzzy controller prototype was integrated with a programmable knowledge base where the rule memory and the blocks that generate the variables' membership functions are implemented with RAM's. The prototype is a three-input one-output fuzzy controller that can employ up to 8 membership functions with an overlapping degree of 2 and a resolution of 6 bits. The rule base memory is implemented as a RAM of 512x6 bits, thus allowing to define a different consequent for each rule. The defuzzifier block has been optimized for the Fuzzy Mean method.

The circuit layout was obtained from the VHDL description provided by the *xfvhdl* tool by using the Mentor Graphics and Cadence design environments. The controller, whose die photograph is shown in Fig. 7, was integrated in a 1.0 µm CMOS technology. Its silicon area is 14.36 mm^2.

Finally, the behavior of the real plant controlled by the PCB that contains the fuzzy integrated circuit can be monitoring by using again the *xflab* utility. The maximum clock frequency measured at laboratory for this IC controller was 27 Mhz, providing an inference speed over 3 MFLIPs.

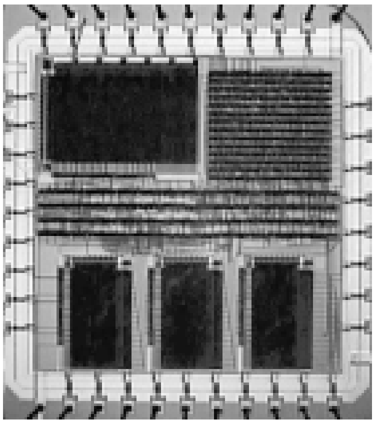


Fig. 7. Fuzzy controller IC implementation.

4. Conclusions

A design methodology for application specific fuzzy controllers has been presented in this paper. The key points of this methodology are the choice of an efficient hardware architecture for fuzzy controllers, and the use of a development environment that eases and speeds-up the different stages of description, verification and synthesis in the design process.

References

[1] A. Costa, A. De Gloria, P. Faraboshi, A. Pagni, G. Rizzotto. "Hardware Solutions for Fuzzy Control". *Proc. of the IEEE*, vol. 83, n. 3, pp. 422-434, Mar. 1995.

[2] D. L. Hung. "Dedicated Digital Fuzzy Hardware". *IEEE Micro*, vol. 15, n. 4, pp. 31-39, Aug. 1995.

[3] D. R. López, C. J. Jiménez, I. Baturone, A. Barriga, S. Sánchez-Solano, "Xfuzzy: A Design Environment for Fuzzy Systems", *Proc. seventh IEEE Int. Conf. on Fuzzy Systems*, Anchorage, May 1998.

[4] D. R. López, S. Sánchez-Solano, A. Barriga: "XFL: A Fuzzy Logic Systems Language", *Proc. sixth IEEE Int. Conf. on Fuzzy Systems*, vol. 3, pp. 1585-1591, Barcelona, 1997.

[5] C. J. Jiménez, S. Sánchez-Solano, A. Barriga, "Hardware Implementation of a General Purpose Fuzzy Controller", *Proc. IFSA World Congress*, vol. 2, pp. 185-188, Sao Paulo, 1995.

[6] E. Lago, C. J. Jiménez, D. R. López, S. Sánchez-Solano, A. Barriga, "XFVHDL: A Tool for the Synthesis of Fuzzy Logic Controllers", *Proc. Design, Automation and Test in Europe*, pp. 102-107, Paris, Feb. 1998.

[7] E. Lago, M. A. Hinojosa, C. J. Jiménez, A. Barriga, S. Sánchez-Solano. "FPGA Implementation of Fuzzy Controllers". *Proc. XII Conference on Design of Circuits and Integrated Systems*, pp. 715-720, Sevilla, Nov. 1997.

[8] S. Sánchez-Solano, A. Barriga, C. J. Jiménez, J. L. Huertas, "Design and Application of Digital Fuzzy Controllers", *Proc. sixth IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 869-874, Barcelona, 1997.