Available online at www.sciencedirect.com
ScienceDirect
Int. J. Electron. Commun. (AEÜ) 64 (2010) 433–442
International Journal AEÜ of Electronics and Communications
www.elsevier.de/aeue

Design of a general propose neuro-fuzzy controller by using modified adaptive-network-based fuzzy inference system

Alireza Peymanfar*, Abdollah Khoei, Kheyrollah Hadidi

Microelectronic Research Laboratory, Urmia University, Beheshti Avenue, Urmia 57159, Iran

Received 5 April 2008; accepted 27 February 2009

Abstract

This paper presents a new general purpose neuro-fuzzy controller to realize adaptive-network-based fuzzy inference system (ANFIS) architecture. A two input, single output, 16 rules ANFIS architecture is designed in 0.35 μm standard CMOS technology. This controller can also be used as a standard (Mamdani) type fuzzy logic controller (FLC) having bell-shaped input and singleton output membership functions. Mixed mode realization of the circuit makes the design programmable and extendable, while having relatively low power consumption. Current mode realization of the rule base and defuzzifier circuits leads to simple and intuitive configurations. For a particular set of programming parameters, simulation results of the controller using HSPICE simulator and level 49 parameters (BSIM3V3), shows an average power consumption of 8 mW and an RMS error of 1.3% compared to ideal results obtained from MATLAB.

Keywords: WTA; Current mode multiplier; CMOS neuro-fuzzy controllers; ANFIS; Learning rule

1. Introduction

The use of fuzzy systems is widespread, mainly in the control field. Most fuzzy microcontrollers presently used in industrial applications are digital implementations. However, fuzzy logic is intrinsically more like the multivalued and continuous analog world than the digital one. In addition, fuzzy systems process this continuous information in a massively parallel way. Furthermore, in many applications the knowledge describing the expected behavior of the system is contained on data clusters. Due to this, the designer has to elaborate the IF–THEN rules according to such data; if the data clusters are too large, it could imply a tremendous effort. Neural networks can learn from data clusters, so it results natural thinking in a methodology which gathers characteristics of both systems, combining explicit knowledge representation of fuzzy logic with the learning capability of neural networks. In this way, the called neuro-fuzzy systems are obtained. Among the various inference methods reported in the literature, the singleton or zero-order Takagi–Sugeno–Kang's (TSK) method is very adequate for hardware implementations. Functionally, the adaptive-network-based fuzzy inference system (ANFIS) architecture is equivalent to a TSK zero order and/or first order fuzzy system [1]. In Ref. [2], ANFIS architecture is discussed and optimized using a new algorithm. This algorithm is suited to use in CMOS circuits. In this paper, design of a two input, single output, 16 rules neuro-fuzzy based on ANFIS architecture is described. It can be programmed as an either first order or zero order TSK fuzzy logic controller (FLC) (which the latter corresponds to a standard type FLC with singleton output membership functions). This controller can realize an input/output mapping based on both human

* Corresponding author.
E-mail addresses: alireza.peymanfar@gmail.com (A. Peymanfar), a.khoei@urmia.ac.ir (A. Khoei), kh.hadidi@urmia.ac.ir (K. Hadidi).

1434-8411/$ - see front matter © 2009 Elsevier GmbH. All rights reserved.
doi:10.1016/j.aeue.2009.02.008

knowledge and stipulated input/output data pairs. Input signals are voltage and output signal is current. Input membership functions are bell-shape that their position can be tuned continuously and their slope are programmed by four digital bits. All blocks except for fuzzifier block of the controller are in current mode so the controller is simple. Each of the blocks has high accuracy, low power consumption and small occupied area. Divider circuit is not used by using modified ANFIS architecture [2], and it reduces the occupied area and power consumption. The block diagram of the controller is described in Section 2, followed by the transistor level design of its blocks in Section 3. In Section 4, simulation results for each block and also the complete controller using HSPICE simulator is presented, and is compared with the results of MATLAB software to show the efficiency of the design. In Section 5, the proposed controller is described systematically. Finally Section 6 concludes the paper.

2. Functional block diagram

The complete block diagram of the proposed neuro-fuzzy controller is shown in Fig. 1. The node functions, in the same layer, are of the same function family as described below [2,3]:

Layer 1: This layer consists of membership function generators (MFG). The universe of discourse of inputs is partitioned into four regions (fuzzy sets) named very small, small, large, very large. These MFGs are bell-shaped functions as shown in Fig. 2:

$$\mu = \frac{Maximum\_Current}{1 + \left(\frac{x-c}{a}\right)^{2b}} \quad (1)$$

where x is input and a, b, c are parameters.

Layer 2: Every node in this layer computes the maximum of incoming signals. Output of each node presents the firing strength of a rule. Because the controller has two inputs and four fuzzy sets for each, total number of rules considered to be $4^2 = 16$ rules.

Layer 3: Each node in this layer is a multiplier. Layer 2 outputs are multiplied with their own singletons to realize consequent part outputs.

Layer 4: The single node in this layer computes the overall output as the summation of all incoming signals. In the next section, a detailed discussion of the blocks and their circuits is presented.

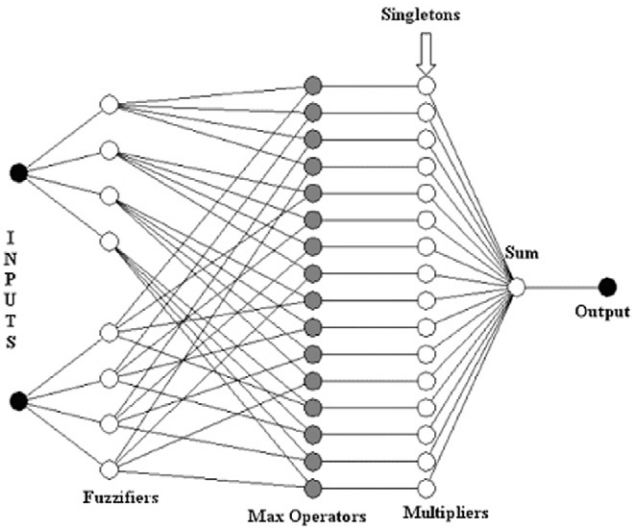

Fig. 1. Complete block diagram of proposed neuro-fuzzy controller.

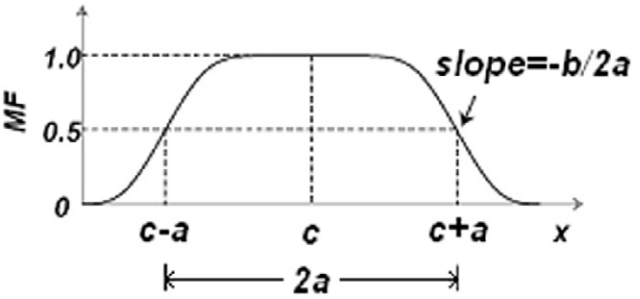

Fig. 2. Physical meanings of the parameters in the bell membership function.

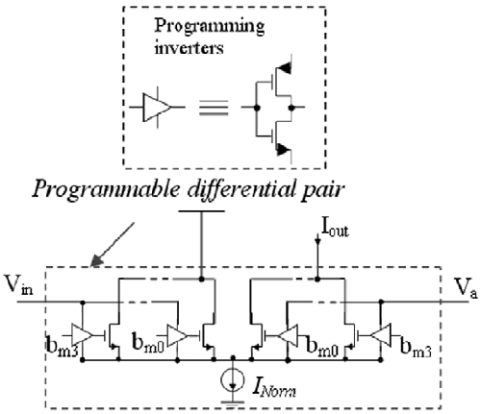

Fig. 3. Conceptual circuit of S-shaped MFG.

3. Circuit design of proposed controller

The design procedure of a neuro-fuzzy controller begins with the selection of input voltages and current of devices. There is a trade off for these: lower current ranges improve power consumption and chip size, but suffer from device mismatching, error caused by leakage current of transistors, and low speed of circuit. A maximum current of 5–50 μA is typical [4]. In the proposed controller, the universe of