Hardware Architecture for an Interval Type-2 Fuzzy Processor

Mamta Khosla1, R K Sarin2 and Moin Uddin3

1,2Department of Electronics and Communication Engineering
Dr B R Ambedkar National Institute of Technology
Jalandhar- 144011, India

3Delhi Technological University, Delhi – 110042

khoslam@nitj.ac.in, sarinrk@nitj.ac.in, prof_moin@yahoo.com

Abstract

This paper presents hardware architecture for an Interval Type-2 fuzzy processor. The architecture is based on the use of two Type-1 Fuzzy Logic Systems (T1 FLS) and it suggests that the existing techniques for implementing T1 FLS can be employed for realizing Type-2 Fuzzy Logic Systems (T2 FLS). The paper also presents a survey of the various realizations for T1 FLS and summarizes them giving their benefits and limitations.

Key Words: analog modules, rule-by-rule architecture, Interval type-2 fuzzy logic systems, Type-2 fuzzy logic systems, type-reduction.

1. Introduction

Hardware implementation of T1 FLSs, now known as conventional fuzzy logic systems is a well-known area [1]. The approaches for implementing these systems cover technologies like microcontrollers, FPGAs among others [2]-[13]. On the other hand, the hardware realization of Type-2 Fuzzy Logic Systems (T2 FLSs) is a relatively nascent research area.

T1 FLSs, which use precise and crisp Type-1 fuzzy Sets (T1 FSs) cannot handle the uncertainties associated with most of the real world applicationslike noisy training data, noisy measurements and multiple meanings of linguistic categories. Hence, they result in poor control and inefficient performance and may require frequent redesigning or tuning the T1 FLS so that it can deal with the various uncertainties [14]. However, T2 FLSs, which use Type-2 Fuzzy Sets (T2 FSs) characterized by fuzzy MFs, have an additional third dimension. This third dimension and footprint of uncertainty (FOU) provide an additional degree of freedom for T2 FLSs to directly model and handle uncertainties [15]. Thus, T2 FLSs are expected to perform better than their traditional counter parts.

This fact and the necessity of facing real-world applications have encouraged the development of hardware implementations of T2 FLSs. A few implementations reported in literature have been around microcontrollers, FPGAs etc. [16]-[20]. Particularly, these implementations have been focused on Interval Type-2 Fuzzy Logic Systems (IT2 FLSs) since they are computationally simpler than general T2 FLS. These implementations are based on the general architecture of a T2 FLS shown in Figure 1, which requires type reduction in addition to the processing stages of a T1 FLS.

In this paper, hardware architecture for an IT2 FLS is presented that uses two T1 FLSs for its implementation. More precisely, an IT2 FLS is approximated with the average of two T1 FLSs and thus it avoids the complications and intensive computations required for type reduction.

The paper is organized as follows: Section II discusses briefly about IT2 Fuzzy Sets (IT2 FS). Section III presents the hardware architecture for an IT2 fuzzy processor and Section IV summarizes the various realizations for T1 FLS along with their advantages and limitations. The conclusions have been drawn in Section V.

![Figure 1. Type-2 FLS][21]
2. IT2 fuzzy sets and footprint of uncertainty (FOU)

A T2 FS, denoted by $\tilde{A}$, can be expressed mathematically in the following form as in (1)

$$\tilde{A} = \{(x,u), \mu_{\tilde{A}}(x,u) \mid \forall x \in X, J_x \subseteq [0,1]\}$$

(1)

Where, $\mu_{\tilde{A}}(x,u)$ is the T2 MF, $u$ being the third dimension of the MF, defined by secondary membership grades,

$x \in X$ and $u \in J_x \subseteq [0,1]$

The MFs of a T2 FS have an FOU, which represents the uncertainties in the shape and position of a T1 FS. It is the union of all primary memberships [15] as in (2)

$$\text{FOU}(\tilde{A}) = \bigcup_{x \in X} J_x$$

(2)

All the embedded FSs of FOU are T1 FSs and their union covers the entire FOU.

IT2 is a special case of a T2 FS where all the secondary membership grades equal one. IT2 FS, shown in Figure 2, is completely characterized by its 2-D FOU that is bound by a Lower MF (LMF) and an Upper MF (UMF) both of which are T1 MFs. IT2 FSs are the most widely used T2 FSs to date used in almost all applications because all calculations are easy to perform. Because of the computational complexity of using a general T2 FLS, most designers only use IT2 FSs in a T2 FLS, the result being an IT2 FLS. LMF and UMF together are popularly used in most of research papers to represent IT2 FLSs [22].

3. Hardware architecture for IT2 fuzzy processor

In spite of the advantages offered by T2FLSs, one major problem that may hinder their use in real time applications is their high computational cost. Type reduction, which is used to convert T2FSs into T1FSs so that they can be processed by the defuzzifier to give a crisp output, is computationally very intensive, especially when there are many MFs and the rule base is large [21].

The architecture of IT2 fuzzy processor presented here reduces the computational burden while preserving the advantages of T2FLSs. This architecture is based on the use of two T1 FLSs to emulate an IT2 Fuzzy System. The first T1 FLS is constructed using LMFs and the second one with the UMFs so as to emulate the FOU in an IT2 Fuzzy System. The three processing stages of the fuzzy processor viz. fuzzification, fuzzy inference and defuzzification are done as traditionally for two T1 FLSs and the outputs are averaged as shown in the Figure 3 [23].

![Figure 2. FOU of an IT2 FS](image)

![Figure 3. Architecture for T2 FLS based on two T1 FLSs](image)

4. Summary of the existing T1 FLS realizations

The architecture of IT2 fuzzy processor presented in section III suggests that the existing T1 design approaches reported in literature can be adopted to realize an IT2 FLS. This section summarizes the various realizations of T1 FLS suggested by researchers. Broadly these realizations fall in three categories.
4.1. Analog implementation

Sanz[24] has suggested the use of op-amps and some commonly available electronic components viz. resistors and diodes for implementing a conventional FLC.

Figure 4. T1 FLS based on rule-by-rule architecture

Different analog modules for fuzzification such as membership function generating circuits (MFCs), modules related to inference engine, such as fuzzy operator implementation circuits and defuzzification circuits are suggested. Further these modules can be organized by the designer to implement FLC specific to his particular application. Their modules can realize the four basic membership functions (MFs) viz. Z, triangular/trapezoidal and S, and the fuzzy operators MIN and MAX. This implementation follows rule-by-rule architecture and provides a high inference speed and hence good performance. This approach is capable of accommodating any number of input and output variables in contrast to fuzzy chips, where the number of inputs and outputs is fixed. Further, this approach is inexpensive with small development time as it uses commonly available electronic components. Limitations of this approach are lack of flexibility and excessive wiring.

A few microelectronic realizations of the rule-by-rule architecture of an FLC using analog techniques reported in literature are [25] – [31]. Yamakawa [25, 26] carried out the first analog realization. These ICs implement MIN-MAX Mamdani’s method with the Center-of-–Area (COA) defuzzification method. Hence, fuzzy sets are used to represent both antecedents and the consequents of the rules. In their work, MFs that define the antecedent fuzzy sets are generated by membership function circuit (MFCs) which are built around bipolar transistors along with resistors, diodes, and programmable switches. And the MFs that define the consequent fuzzy sets are generated by membership function generators (MFGs) which are memory based i.e. they store the membership degrees corresponding to all the discrete points within the universe of discourse. The architecture proposed by them is shown in Figure 5. The realizations described in [29, 30] are programmable realizations that contain memory circuitry to store the programmable parameters of the MFCs and MFGs.

Figure 5. Parallel architecture proposed by Yamakawa [25, 26]

Memory approach for realizing the MFC fulfills the speed requirement of a fuzzy processor, but is complex for VLSI realization and power consuming as well [32]. Many recent researches on the design of analog FLC have reported arithmetic MFCs that generate membership function with their transfer curve. Three modes of operation are possible with these circuits: current mode, voltage mode, and transconductance mode. Ota [33] has used the voltage mode approach, whereas current mode MFCs have been reported in [34] – [38]. The basic building blocks of this type of MFCs are operational transconductance amplifiers (OTA). An OTA is used as a voltage to current converter. The position of the membership function is controlled by the applied voltage at the gate terminals of the differential transistor pairs of the OTA, and the slope is adjusted by the sizes of the transistor pairs. Advantages of current mode MFCs with respect to the voltage mode MFCs are that they have high speed, low power consumption, large dynamic range, reduced chip size and they allow natural form of addition and subtraction. An MFC based on this approach is shown in Figure 6.

Figure 6. Current mode MFC

Advantages of analog realizations are their high speed and small area and low power consumption. Also these
realizations follow the rule-by-rule architecture, and hence process the rules in parallel so that these response times do not depend on the number of rules. Furthermore, in most of the applications where the inputs are analog, analog fuzzy chips are preferred because digital fuzzy chips require A/D and D/A converters.

4.2. Digital implementation

In comparison to analog counterpart, digital hardware offers the advantage of being more accurate and robust. Moreover, significant advances in the digital domain have made the digital system design, verification and implementation process more efficient than that of analog design. Many fuzzy systems realizations are based on the use of general-purpose microprocessor/microcontroller for fuzzy inference. This approach provides the highest level of flexibility and is very economical but has slow inference. Many real-time applications require high inference speed that can only be provided by application specific fuzzy hardware. Digital fuzzy hardware falls into two main categories viz. fuzzy processors and dedicated fuzzy hardware. A fuzzy processor is an application specific instruction processor with a set of special instructions to support certain fuzzy logic operations that provides higher performance than a standard general-purpose processor and is flexible as well. On the other hand, fuzzy dedicated hardware yields the highest performance at the expense of low flexibility and high cost.

Some initial digital fuzzy integrated circuits employed architectures that provide a data path for each rule [39, 40, 41]. This provides a very high inference speed. Other two architectures for digital fuzzy ICs reported in literature are: sequential rule processing architecture and active rule-driven architecture. Some of the work based on the sequential rule-processing architecture has been reported in [42, 43, 44]. The inference speed for this architecture is dependent on the number of rules since the processing of rules is sequential. An active rule-driven architecture only needs the circuitry to identify and process the active rules. A few fuzzy implementations in digital domain based on this architecture can be found in [45, 46].

Another very good option for implementing FLCs has been on user programmable devices like field programmable gate array (FPGA) that offer the flexibility of general-purpose processors and performance of dedicated fuzzy hardware with shorter development time. Some of the implementations of FLC on FPGAs reported are [13, 47, 48].

4.3. Mixed signal

Mixed signal approach can capture the best of both analog and digital implementations. The analog circuitry provides the low power, high speed and small chip area, whereas the digital part offers programmability. The CMOS mixed signal realizations reported by [49] - [53] have used CMOS circuits realized with current mode continuous time techniques, which provide high functional efficiency in realizing the basic mathematical operations required in the different functional blocks of the FLS like addition, subtraction. Further, in [49] and [50], current mode D/A converters are used to implement the scaling operation and to program antecedent and consequent parameters. Whereas, in [51], current mode A/D converters are used to complete and implement the defuzzifier block. This FLC has mixed-signal (analog and digital) inputs and digital outputs. In [52] and [53], a digitally programmable analog Fuzzy Logic Controller (FLC) is designed, where the input and output signals are processed in the analog domain and the parameters of the controller are stored in a built-in digital memory.

5. Conclusions

We have suggested hardware architecture for an IT2 fuzzy processor that is based on the use of two T1 FLSs. The architecture can exploit the existing T1 design approaches for realizing T2 FLS.

To the best of our knowledge, there is no report of a parallel hardware implementation of an IT2 FLS using analog modules built around op-amps, diodes, and resistors. Neither is any implementation around CMOS reported in the literature. The authors are currently working on realizing these two implementation approaches for designing an IT2 fuzzy processor based on the hardware architecture.

6. References


