FUZZY LOGIC SYSTOLIC ARRAY FOR REAL-TIME APPROXIMATE REASONING

Jia-Yuan Han
Department of Electrical Engineering
Southern Illinois University
Carbondale, Illinois 62901-6603

Jia-Lian Han
Center for Polymer Physics
Boston University
Boston, Massachusetts 02215

Abstract

In this paper fuzzy logic is introduced for approximate reasoning, which is a better model for human beings reasoning. Application of fuzzy logic to rule-based expert systems makes the systems more intelligent like human beings. In order to realize the approximate reasoning in real-time, we propose two systolic arrays for the fuzzy logic in rule-based expert systems. Parallel and pipeline processing is realized in the systolic arrays for fast, approximate reasoning. The step-processors in the systolic arrays are also developed.

I. Introduction

Artificial intelligence is a fast growing interdisciplinary science and has found its applications in numerous fields ranging from computer science and engineering to social sciences and medical practice. Rule-based expert systems and their applications have shown great interest and rapid development. The development of hardware and software support for artificial intelligence, especially for rule-based expert systems, is an important topic in computer engineering [1,2]. With the development and applications of intelligent control and artificial intelligence, many intelligent algorithms and expert systems have been developed. When the searching space is very large in the intelligent algorithms and systems, the computation time increases tremendously. To reduce the searching space, fuzzy inference may be applied to the intelligent systems [3]. Various parallel and pipeline processing architectures have been proposed to reduce the computation time to realize intelligent algorithms in real-time [4].

Fuzzy subset theory was introduced by Zadeh in 1965 [5]. It is a powerful method to formulate fuzzy environment and human being's reasoning [6]. With the development of artificial intelligence, fuzzy logic finds it way to formulate approximate reasoning for intelligent systems. The application of fuzzy logic in the intelligent systems not only makes the system's reasoning and decision making more like human beings, but also reduces the searching space in the traditional logic in the systems. The detail application of fuzzy-logic to expert systems can be found in [3,6, and 7]. The hardware implementation of a special approximate reasoning is described in [8].

The rapid development of VLSI technology makes hardware inexpensive. Various powerful microprocessors and a large amount of memory cells have been fabricated in single chips. Systolic array processors are the result of advances in VLSI technology and its applications that require extensive throughput. Since Kung and Leiserson [9] introduced the concept of the systolic array, much research has been done on this topic. The application of systolic arrays ranges from matrix arithmetic computation, non-numerical computations, to signal and image processing and pattern recognition [2]. Parallel and pipeline computing takes place along all dimensions of a systolic array and results in very high computational throughput.

In this paper, a fuzzy-logic systolic array processor is proposed for approximate reasoning in rule-based expert systems. The aim is to speed up approximate reasoning by applying parallel and pipeline processing in the systolic architecture.

The paper is organized in the following way. First, rule-based expert systems are addressed from a logic operation point of view. The simple logic structures realized in this paper are discussed in detail. Second, basic fuzzy logic and approximate reasoning are introduced for further discussion. In Section 4, the logic architectures of the fuzzy inference are addressed in detail. Finally, systolic array architectures and step-processors for the fuzzy inference are developed.

II. Rule-Based Expert Systems

The inference in expert systems is based on rules supplied by human experts. A rule is a conditional statement expressed by If-Then or If-Then-Else form. Suppose that a rule is stated as "if A then B" denoted by A=> B. If an observation is that A is true, according to the rule, we conclude that B is true. It can be expressed as (A=>(A=>B)=>B). For example, the following rules are used in medical practice.

(1) IF organism is streptococcus OR bacteroids THEN penicillin is indicated.

(2) IF penicillin is indicated AND patient allergies are unknown THEN ask about allergy to penicillin.

0-8186-2012-9/89/0000/0628$01.00

(3) IF penicillin is indicated AND NOT allergic to penicillin THEN prescribe penicillin.

This kind of rule-based inference is called exact reasoning.

There are various hierarchy structures in expert systems. In this paper, the simplest single-level "If-Then-Else" structure is discussed. To simplify the discussion, single condition and single action in each conditional statement expressed by If-Then-Else form are considered. The expert systems considered here are described as follows.

Expert System 1

Rule 1: If x is A_1, then y is B_1.

Rule 2: If x is A_2, then y is B_2.

....

Rule n: If x is A_n, then y is B_n.

In this system, every rule is a conditional statement which may or may not be related with other statements. These rules are completely parallel. Therefore, the action of the system depends on the effect of all the rules. The relationship among the rules is AND.

Expert System 2

Rule 1: If x is A_1, then y is B_1

Rule 2: Else if x is A_2, then y is B_2

Rule 3: Else if x is A_3, then y is B_3

....

Rule n-1: Else if x is A_{n-1}, then y is B_{n-1}

Rule n: Else y is B_n.

In this system, every rule is a conditional statement which is related with the previous statements. That is, Rule i will function if and only if Rule 1,2,..., i-2 and Rule i-1 do not function. One can rewrite Expert System 2 as follows.

Expert System 2'

Rule 1: If x is A_1, then y is B_1

Rule 2: If x is $\neg A_1$ and x is A_2, the y is B_2.

....

Rule n-1: If x is $\neg A_1$ and $\neg A_2$,...and $\neg A_{n-2}$, and x is A_{n-1}, then y is B_{n-1}

Rule n: If x is $\neg A_1$ and $\neg A_2$...and $\neg A_{n-1}$, then y is B_n

Each statement in this system is a conditional statement which is independent with other statements. That is, one and only one statement functions among all the statements. The relationship among the rules is OR.

III. Fuzzy Logic and Approximate Reasoning

The reasoning discussed in Section 2 is the exact reasoning, because all of the variables are defined by crispy sets and there is no ambiguity among the concept. However, most concepts in practice are described in human language and represented by imprecise knowledge. For example,

If
(1) The temperature of the room is high
(2) The humidity of the room is OK
(3) The electricity load of the lab is OK, then turn the air-conditioner knob slightly to the cold.

The term high, OK, and slightly in the rule are not precisely defined, say, the temperature of 95° F can be though as high, the temperature of 90° F can also be high. Of course, the degrees of highness are different for 95° F and 90° F. This kind of inference is called inexact reasoning (approximate reasoning).

Due to the pioneer work of Zadeh [5], fuzzy set theory and its application were widely developed. It was found that fuzzy logic and its linguistic approach are good methods for approximate reasoning. The detail discussion of fuzzy set, fuzzy logic, fuzzy linguistics, and approximate reasoning can be found in [6]. Only the concepts needed for a hardware design are presented here.

If A is a finite subset of a universal set U,

$$A = \{u_1,...,u_n\} \subset U \tag{1}$$

A finite fuzzy subset A of U is a set of ordered pairs

$$A = \{u_j, \mu_A(u_j)\}, \quad u_j \in U \tag{2}$$

where the $\mu_A(u_j)$ is the membership function and $0 \le \mu_A(u_j) \le 1$.

A fuzzy relation R from A to B is a fuzzy subset of the Cartesian product U x V, where $A \subset U$ and $B \subset V$. The conditional statement, "If X is A then Y is B," is represented by the fuzzy relation R and defined as follows:

$$\mu_R(u,v) = \min(\mu_A(u), \mu_B(v)), \quad u \in U \text{ and } v \in V \tag{3}$$

Fuzzy Logic is an extension of Boolean logic. The basic logic operations in Boolean logic are AND, OR, NOT and implied, that is, $A \wedge B$, $A \vee B$, A, and A=>B. The fuzzy logic operations are defined as follows. Let the numerical truth values be $v(A)$ and $v(B)$.

$$v(A) \wedge v(B) = v(A \text{ AND } B) = (v, \min(\mu_A(v), \mu_B(v)) \tag{4}$$

$$v(A) \vee v(B) = v(A \text{ OR } B) = (v, \max(\mu_A(v), \mu_B(v))) \tag{5}$$

$$\neg v(A) = v(\text{NOT } A) = (v, 1 - \mu_A(v)) \quad (6)$$

$$v(A) \Rightarrow v(B) = v(A \Rightarrow B) = (v, \max(1 - \mu_A(v), \mu_B(v))) \quad (7)$$

Compositional Rule: If R is a fuzzy relation from U to V, and x is a fuzzy subset of U, then the fuzzy subset y of V induced by x is denoted by

$$y \equiv x \circ R \quad (8)$$

and defined as follows

$$\mu_y(v) = \max_{u \in U} \min(\mu_x(u), \mu_R(u,v)) \quad (9)$$

IV. Logic Architecture of the Fuzzy Inference

In order to realize the rule-based expert systems in fuzzy logic, one has to convert the logic operations to fuzzy-subset operations, such as, min's and max's. Further analysis of the fuzzy-subset operations is necessary for mapping the operations onto systolic array processors. In this section, the two expert systems discussed in Section 2 are addressed and the algorithms are decomposed for systolic operations.

Fuzzy Expert System 1

Suppose that $A_i (i=1,\ldots,n)$ is a fuzzy-subset of U and $B_i (i=1,2,\ldots,n)$ is a fuzzy-subset of V. A fuzzy relation is defined by rules as Expert System 1. The overall relation R is denoted and defined as

$$R = \bigwedge_i R_i = \min_i f_{\rightarrow}(\mu_{A_i}(u), \mu_{B_i}(v)) \quad i=1,\ldots,n. \quad (10)$$

where $f_{\rightarrow}(\mu_{A_i}(u), \mu_{B_i}(v)) = \mu_{A_i \rightarrow B_i}(u,v)$ represents the fuzzy relation "If A_i then B_i".

Given an observation A' and rule R_i, the action B_i' is inferred and defined as

$$B_i' = A' \circ R_i, \quad A' \varepsilon U, B_i' \varepsilon V \text{ and } R_i \subset U \times V \quad (11)$$

and its membership function is expressed as

$$\mu_{B_i'}(v) = \max_{u \varepsilon U} \min(\mu_{A'}(u), \mu_{R_i}(u,v))$$

$$= \max_{u \varepsilon U} \min(\mu_{A'}(u), \min_{\substack{u \varepsilon U \\ v \varepsilon V}} (\mu_{A_i}(u), \mu_{B_i}(v))) \quad (12)$$

The overall decision B' is determined by $B_1', B_2', \ldots, B_n'$, that is

$$B' = \bigwedge_i B_i' \quad (13)$$

$$\mu_{B'}(v) = \min_i \mu_{B_i'}(v) \quad (14)$$

Fuzzy Expert System 2

Suppose that $A_i (i=1,2,\ldots,n)$ is a fuzzy-subset of U and $B_i (i=1,2,\ldots,n)$ is a fuzzy-subset of V. A fuzzy relation is defined by rules as Expert System 2'. The overall relation R is denoted and defined as

$$R = \bigvee_i R_i = \max_i f_{\rightarrow}(\mu_{C_i}(u), \mu_{B_i}(v)), \quad i=1,\ldots,n \quad (15)$$

where $f_{\rightarrow}(\mu_{C_i}(u), \mu_{B_i}(v)) = \mu_{C_i \rightarrow B_i}(u,v)$ represents the fuzzy relation "If C_i then B_i", where $C_i = D_{i-1} \wedge A_i$, $D_i = D_{i-1} \wedge \neg A_i$, $D_1 = \neg A_1$.

Given an observation A' and a rule R_i, the action B_i' is inferred and defined as

$$B_i' = A' \circ R_i, \quad A' \varepsilon U, B_i \varepsilon V, R_i \subset U \times V \quad (16)$$

and all the membership functions are expressed as

$$\mu_{B_i'}(v) = \max_{u \varepsilon U} \min(\mu_{A'}(u), \mu_{R_i}(u,v))) \quad (17)$$

$$\mu_{R_i'}(u,v) = \min_{\substack{u \varepsilon U \\ v \varepsilon V}} (\mu_{C_i}(u), \mu_{B_i}(v)) \quad (18)$$

$$\mu_{C_i}(u) = \min(\mu_{D_{i-1}}(u), \mu_{A_i}(u)) \quad (19)$$

$$\mu_{D_{i-1}}(u) = \min(\mu_{D_{i-2}}(u), 1 - \mu_{A_{i-1}}(u)) \quad (20)$$

and $$\mu_{D_1}(u) = 1 - \mu_{A_1}(u) \quad (21)$$

The overall decision B' is determined by $B_1', B_2', \ldots, B_n'$, that is

$$B' = \bigvee_i B_i' \quad (22)$$

$$\mu_{B'}(v) = \max_i \mu_{B_i'}(v) \quad (23)$$

V. Systolic Array Architecture for Expert Systems

Many techniques to map computer algorithms onto various systolic array processors have been developed [10]. The essential step is partitioning an algorithm to explore the possible parallel and pipeline processing for systolic operations. In this section, the logic architectures in Fuzzy Expert System 1 and 2 are realized in linear systolic arrays.

Systolic Array for Fuzzy Expert System 1

The operations for the fuzzy logic are shown in Equations 12 and 14. From Equation 12, one can use parallel processing to find

$\mu_{R_i}(u,v) = \min_{u \in U, v \in V}(\mu_{A_i}(u), \mu_{B_i}(v))$ for a fixed v. The inputs are $\mu_{B_i}(v)$ for a fixed v and $\mu_{A_i}(u)$ for all the u's. The outputs are $\mu_{R_i}(u,v)$ for a fixed v and all the u's. Then the max-min operation can be completed among all the u's. For Equation 14, one can rewrite it as

$$\mu^{i}_{B'}(v) = \min(\mu^{i-1}_{B'}(v), \mu_{B'_i}(v)) \qquad (24)$$

Then one can develop a step-processor for a systolic array processor to realize Fuzzy Expert System 1 (Figure 1). The inputs are $\mu_{A'}(u)$, $\mu_{A_i}(u)$, $\mu_{B_i}(u)(v_k)$ and $\mu^{i-1}_{B'}(v_k)$ and the outputs are $\mu_{A'}(u)$ and $\mu^{i}_{B'}(v_k)$. The $\mu_{A'}(u)$ passes the step-processor and is used by each step-processor. The $\mu_{B'_i}(v_k)$ is yielded from the $\mu_{A'}(u)$, $\mu_{A_i}(u)$ and $\mu_{B_i}(v_k)$ by applying Equation 12. Finally, the output $\mu^{i}_{B'}(v_k)$ is obtained from Equation 24.

The architecture of the proposed linear systolic array is shown in Figure 2. The sequences to feed $\mu_{B_i}(v_k)$ are also shown in the figure. After n steps, the resulting $\mu_{B'}(v_1)$ appears at the output $\mu^{n}_{B'}(v)$. The total inference takes n + m steps, where n is the number of rules and m is the number of samples of variable v.

<u>Systolic Array for Fuzzy Expert System 2</u>

The operations for the fuzzy logic in the second expert system are shown in Equations 17, 18, 19, 20, 21, and 23. The major differences of System 1 and System 2 are (1) instead of min-operation in Equation 14, the operation in Equation 23 is max, and (2) instead of a fuzzy relation $\mu_{R_i}(u,v) = \min_{u \in U, v \in V}(\mu_{A_i}(u), \mu_{B_i}(v))$ in System 1, the fuzzy relation $\mu_{R_i}(u,v)$ in System 2 is expressed by Equation 18, where the $\mu_{C_i}(u)$ not only depends on $\mu_{A_i}(u)$, but also depends on $\mu_{D_{i-1}}(u)$ from the output of the previous step-processor. Similar to the System 1, one can rewrite Equation 23 as

$$\mu^{i}_{B'}(v) = \max(\mu^{i-1}_{B'}(v), \mu_{B'_i}(v)) \qquad (25)$$

The step-processor for Systolic Array Processor 2 to realize Fuzzy Expert System 2 is shown in Figure 3. The inputs are $\mu_{A_i}(u)$, $\mu_{B_i}(v_k)$, $\mu_{A'}(u)$, $\mu_{D_{i-1}}(u)$, and $\mu^{i-1}_{B'}(v_k)$ where the last three are the outputs of the previous step-processor. The outputs of the step-processor are $\mu_{A'}(u)$, $\mu_{D_i}(u)$, and $\mu^{i}_{B'}(v_k)$, where the $\mu_{A'}(u)$ passes the step-processor, the $\mu_{D_i}(u)$ are updated by min-operation with $1 - \mu_{A_i}(u)$, and the resulting $\mu^{i}_{B'}(v_k)$ is updated in this step-processor. The operations within the step-processor are described by Equation 17, 18, 19, 20, and 25.

The architecture of the proposed linear systolic array processor is shown in Figure 4. The feeding sequences of $\mu_{B_i}(v_k)$ are also shown in the figure. The total steps for computing a single $\mu_{B'}(v_k)$ are n. If there are m samples for variable v, the total steps for computing $\mu_{B'}(v_k)$ (k = 1,...,m) are n + m. Even the numbers of steps for both systolic arrays are the same, the lengths for each step are different. The step length of Systolic Array Processor 2 is longer since there is one more operation expressed by Equation 19.

VI. Conclusion

After discussing rule-based expert systems and approximate reasoning in fuzzy-logic, we develop the logic architectures for two fuzzy-logic expert systems. Then we map the logic architectures onto two linear systolic arrays and develop the step-processors for the systolic arrays. The results show that the fuzzy-logic rule-based expert systems can be realized in systolic arrays with high parallelism and throughput. Therefore, real-time realization of expert systems is possible. Further research can be conducted to various approximate reasoning, various hierarchy reasoning structure, and various kinds of systolic array architecture.

References

[1] J. A. B. Fortes and B. W. Wah, "Systolic Arrays - From Concepts to Implementation," Computer, July 1987, pp. 12-17.

[2] B. W. Wah, "New Computers for Artificial Intelligence Processing," Computer, Jan. 1987, pp. 10-15.

[3] E. Sandez and L. A. Zadeh, <u>Approximate Reasoning in Intelligent Systems, Decision and Control</u>, Pergamon Press, Oxford, England, 1987.

[4] B. Wah and G. -J. Li, <u>Computers for Artificial Intelligence Applications</u>, IEEE Computer Society Press, New York, 1986.

[5] L. A. Zadeh, "The Concept of a Linguistic Variable and Its Application to Approximate Reasoning," Memo ERL-M411, Berkley, Oct. 1973.

[6] H. J. Zimmermann, Fuzzy Set Theory and Its Applications, Kluwer-Nijhoff Publishing, Boston, 1985.

[7] C. V. Negoita, Expert Systems and Fuzzy Systems, Benjamin/Cummings Publishing, Monlo Park, CA, 1985.

[8] M. Togai and H. Watanabe, "Expert Systems on a Chip: An Engine for Real-time Approximate Reasoning," IEEE Expert, Fall, 1986, pp. 55-62.

[9] H. T. Kung and C. E. Leiserson, "Algorithms for VLSI Processor Arrays," in Introduction to VLSI Systems, C. A. Mead and L. A. Conway, Eds., Reading, MA, Addison-Wesley, 1980, pp. 271-292.

[10] J. J. Navarro, J. M. Llaberia, and M. Valero, "Partitioning: An Essential Step in Mapping Algorithms into Systolic Array Processor, Computer, July1987, pp. 77-89.

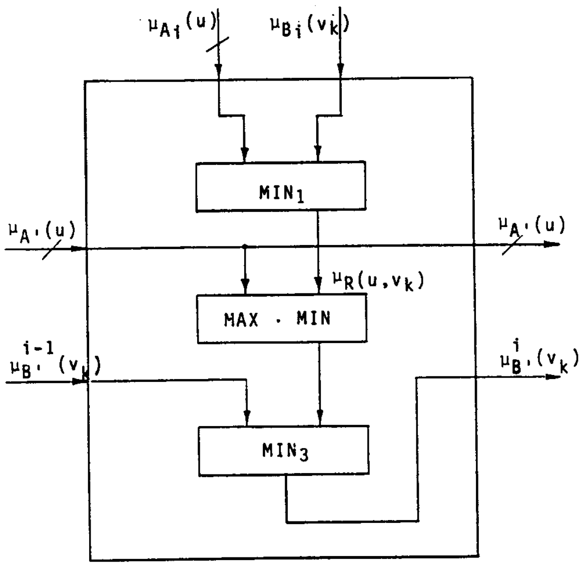

Fig. 1 Step processor for Systolic Array 1

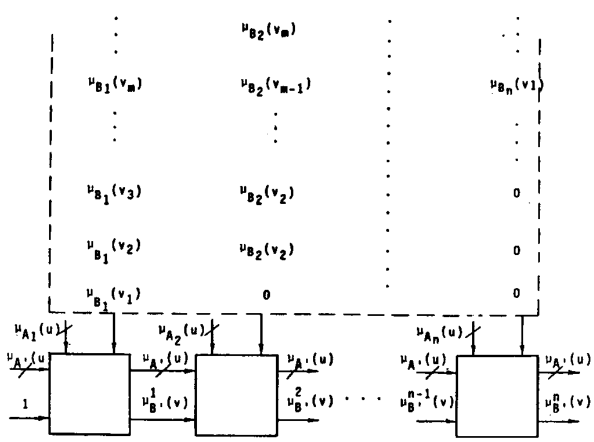

Fig. 2 Systolic Array for Fuzzy System 1

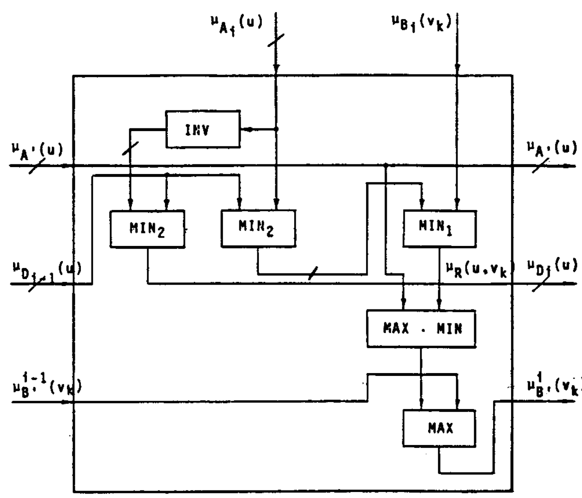

Fig. 3 Step processor for Systolic Array 2

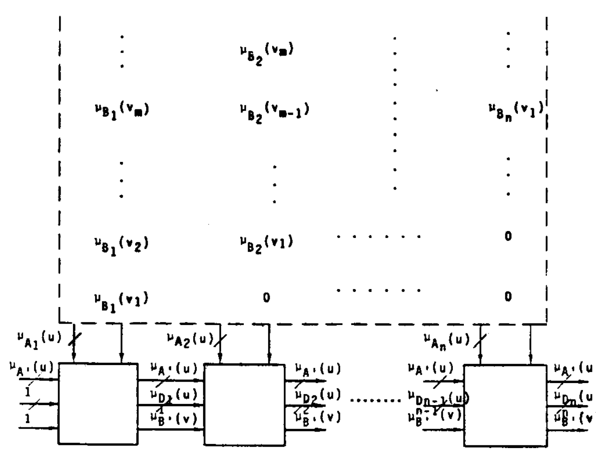

Fig. 4 Systolic Array for Fuzzy Expert System 2