고성능 병렬 퍼지 아키텍처의 설계 및 구현

이 상 구†

요 약

본 논문에서는 Mamdani 방법과 Koczy 방법의 퍼지 추론 알고리즘에 대해서 병렬머신에 적합한 병렬 퍼지 추론 방법을 제안하고, 효율적인 병렬 퍼지 아키텍처를 설계한다. 제안된 아키텍처는 비교적 높은 성능을 갖고, 확장이 용이한 구조로서, 여러개의 FPE(Fuzzy Processing Element), CP(Control Processor), 메모리 모듈, 상호연결망 및 Min 회로로 구성되어 있다. 이러한 구조의 특징은 i번째의 FPE는 i번째의 전건부 및 i번째의 후건부의 처리만을 수행하기 때문에 전건부, 변수들의 처리는 각각 병렬로 수행되고, 후건부의 처리도 또한 각각 병렬로 수행된다. 따라서 프로세서의 활용도가 높아지며, 전건부와 후건부의 변수, 퍼지규칙의 수에 관계없이 쉽게 구성할 수 있다. 이러한 구조는 실시간에 고속추론을 요하는 시스템 또는 전건부와 후건부의 변수가 많은 대규모 전문가 시스템에 사용되어 질 수 있으며, MISO (Multiple-input, Single-output) 시스템보다 MIMO (Multiple-input, Multiple-output) 시스템에 특히 적합하다.

Design and Implementation of High-Performance Parallel Fuzzy Architecture

Sang Gu Lee†

ABSTRACT

In this paper, we present parallelizing techniques for two fuzzy inference methods (Mamdani and Koczy methods) and propose an effective parallel fuzzy architecture, which is suitable for fuzzy information processing. The proposed parallel fuzzy architecture is composed of FPEs, CP, memory modules, interconnection network and Min circuits. This architecture achieves a relatively high performance and it is a generalized cascadable architecture. Here, each FPEi processes only the operations of i-th antecedent and i-th consequent. Then, all of the antecedents in the condition part are executed in parallel, as well as in the consequent part. So processors can be fully utilized and it is easy to configure the design with any number of antecedents, consequents and rules. This architecture can be used in a system requiring a rapid inference time in real-time system and/or a large expert system that has many inference variables in condition part and consequent part. It is especially well suited to the MIMO (Multiple-input, Multiple-output) than the MISO (Multiple-input, Single-output) system.

1. 서 론

퍼지이론은 Zadeh에 의해 제안된 이래 제어분야에 국한되지 않고, 공학분야(엔지니어링 설계, 지능제어, 신호처리, 패턴인식, 이상진단), 사회분야(의사결정, 행동과학, 경제·사회 모델), 자연분야(기상, 인공위성 지도, 생태계, 물리화학적 현상규명) 및 의료분야 등 여러 분야에 걸쳐 많은 응용 및 연구가 진행되고 있다. 퍼지 이론이 많은 분야에 응용되고 있음에도 불구하고,

※ 본 연구는 1998년도 한남대학교 교비 학술연구 조성비 지원에 의하여 수행되었음.
† 정 회 원 : 한남대학교 컴퓨터공학과
논문접수 : 1998년 2월 26일, 심사완료 : 1998년 5월 4일

퍼지이론을 적용하기 위해 반드시 수반되는 전문가의 지식베이스의 규칙들을 쉽게 얻기 어렵고, 퍼지 논리가 0과 1의 이진 논리가 아닌, 0과 1 사이의 실수 연산이 필요하기 때문에 수천 개 이상의 퍼지규칙을 갖는 퍼지 추론을 실시간(real-time)내에 수행하기 쉽지 않으므로, 퍼지 연산이 대규모로 실행되면서 실시간성이 요구되는 시스템이나 대규모의 전문가 시스템 등에서의 실시간 처리에는 아직 해결하여야 할 문제점들이 남아 있다. 또한 지금까지 개발된 대부분의 퍼지 하드웨어들은 AND/OR(Min/Max)의 연산은 병렬로 수행하고 있지만 퍼지규칙들에 대해서는 순차적으로 수행하고 있으므로, 퍼지규칙들에 대해 병렬로 수행할 수 있는 새로운 기법에 대한 연구가 필요하다.

퍼지제어기는 일반적으로 퍼지화 단계(fuzzifier), 퍼지추론 엔진(fuzzy inference engine), 비퍼지화 단계(defuzzifier)의 3부분으로 구성된다. 퍼지화 단계는 입력 변수들의 값을 측정하여 애매하지 않은(crisp한) 입력 값을 퍼지 추론을 행할 수 있는 퍼지집합으로 변환하는 단계이다. 퍼지추론 엔진 단계는 퍼지논리 제어기의 실행부로 퍼지관계와 퍼지논리의 추론규칙을 도입하여 인간의 의사결정 방식과 유사하게 퍼지 제어기로의 출력값을 구해주는 단계이다. 비퍼지화 단계는 퍼지추론에 의한 애매한 결과를 애매하지 않은 하나의 수치로 변환하는 역할을 하며 추론 결과의 해석이라고 말할 수 있다. 위의 3단계 중에서 가장 중요한 부분이 퍼지추론의 단계이며 퍼지 연산 전용의 추론 하드웨어의 개발이 많이 연구되어 왔다. 또한 퍼지 추론 과정을 빠르게 처리할 수 있는 전용 하드웨어의 개발이 세계적으로 많이 진행되고 있는데, 이것은 퍼지추론의 과정이 하드웨어화하기 쉽고, 병렬처리를 통한 추론시간의 감소를 꾀할 수 있는 장점을 갖고 있기 때문이다. 퍼지논리 추론의 전용 하드웨어 개발에 대한 연구는 Yamakawa[19], Togai[15], Watanabe[17]의 연구를 시작으로 일본의 Omron, Oki, Hitachi, 미국의 American NeuraLogix Inc., Togai Infralogic Inc., VLSI Technology Inc., 독일의 Siemens AG 등 여러 곳에서 진행되어, 프로세서 형태의 칩 또는 칩을 장착한 가속보드의 형태로 판매하고 있는 실정이다.

Yamakawa[19]에 의해 개발된 퍼지 하드웨어는 전형적인 analog 방식의 퍼지 추론으로서 퍼지 소속함수의 값이 0과 1들로 표현할 수 있는 digital 값이 아니라, 0V에서 5V 사이의 전압으로 표현되는 방식이다. 현재까지 개발 및 연구중인 퍼지 추론을 위한 하드웨어는 이와 같은 analog 방식보다는 digital 방식의 퍼지추론이 주류를 이루고 있다. Togai, Watanabe[15]가 개발한 근사추론엔진은 digital 방식의 퍼지추론 시스템의 전형적인 모델이다. 이러한 Yamakawa, Togai, Watanabe의 연구를 시작으로 하여 일본에서는 여러 회사에서 퍼지 하드웨어를 개발하기 시작하였다. 일본에서는 1989년에 세계최초의 analog hybrid IC판을, 1990년에는 제어용 디지털 보드를 개발하였으며, 용도별 퍼지 프로세서의 개발도 진행되어 왔다. 종래에는 퍼지추론 프로세서들의 대부분이 특수목적용으로 ASIC 방법에 의해 설계되어 왔다. 이들은 고속으로 동작하지만 일반성이 제한되어 있다. Watanabe[18]는 RISC로 구성되어 있는 퍼지 아키텍처를 설계하였는데, 이의 특징은 퍼지 정보처리를 위하여 RISC 구조에 맞는 명령어 집합을 설계하였고, 퍼지 연산동작들이 벡터 명령으로 수행되도록 하였으며 낮은 가격으로 벡터 명령을 구현할 수 있는 방법을 제시하였다. 그러나 퍼지처리를 위한 스칼라 유닛과 벡터 유닛을 효과적으로 통합하는 방법에 대한 연구는 되어있지 않았다. Yamakawa는 [20]에서 analog 방식으로 되어있는 퍼지 rule chip FP9000과 defuzzifier chip FP9001을 제작하여 발표하였다. 그리고 이들의 analog chip을 digital 회로 및 버스에 연결시키기 위한 인터페이스에 대한 연구도 수행하였다. 이러한 FP 시리즈의 퍼지 프로세서에 의한 설계방법은 하드웨어 구성이 쉽지만, 하드웨어가 고정되어 있어 유연성이 거의 없는 실정이고, single-output 퍼지제어 시스템에 기반을 두어 설계하였다. 이러한 퍼지 chip을 이용하여 Jaramillo-Botero[4]는 일반 PC의 외부 확장 슬롯에 FP시리즈에 호환가능한 디지탈 인터페이스를 설계하여 여러 개의 rule chip을 달아 여러 개의 퍼지 유닛을 구성하였고, 이들의 출력을 모아 하나의 defuzzifier 유닛을 구성하여 PC에서 제어할 수 있는 시스템을 만들었다. 한편 Ungering과 Goser[16]는 퍼지추론 프로세서 FIP와 64비트 마이크로프로세서를 이용하여 64-비트의 퍼지추론 프로세서를 설계하여, 추론시의 speed-up이 약 10배 정도가 된다. 이 시스템의 특징은 퍼지추론은 FIP에 의해 수행되고, 비퍼지화는 마이크로프로세서에 의해 수행된다. FIP의 구조는 IF 부분과 THEN 부분의 모듈로 나누어 구성되어 있으며 FIP의 제어는 호스트 마이크로프로세서에 의해 이루어진다. 또한 Unge-

ring은 1994년 범용의 8-bit 마이크로컨트롤러에 퍼지 기능을 추가시킨 F166을 개발하였으며, Avodadro 등은 1994년에 통상의 RISC 명령이외에 퍼지 응용을 위한 Max 연산과 Min 연산 명령을 추가한 RISC 프로세서 FLORA를 개발하였다. 이들의 특징은 범용의 프로세서에 퍼지 연산 기능을 부가하는 것에 의해, 범용성을 잃지 않고 퍼지 연산의 성능을 향상시킬 수 있는 프로세서에 대한 연구이다. 한편 퍼지 제어 시스템을 효율 좋게 처리하기 위해서는 퍼지 규칙의 후건부 처리 및 비퍼지화 처리의 최적화가 중요하다는 것이 지적되고 있다. 또한 퍼지 추론 연산에 대해서 약 70%의 시간은 데이터의 전송에 걸리므로 퍼지 프로세서의 설계에 있어 이러한 범용 프로세서의 설계상 고려해야 하는 사항을 충분히 염두해야 할 필요성이 있다.

지금까지의 일련의 연구결과 및 연구 흐름을 종합해 보면 퍼지 추론 칩은 여러 가지가 개발되어 있으나 하드웨어가 고정되어 있어서 유연성이나 확장성은 거의 없는 실정이다. 또한 퍼지 추론 시스템은 퍼지 지식베이스에 대해 하나의 규칙을 단위로 하여 추론을 실행한다. 그러므로, 대부분의 퍼지 추론 시스템에서는 규칙들을 해석하는 추론단계에서 순차적으로 수행이 된다. 그러나, 고속으로 퍼지 추론을 요구하는 시스템이나 대규모의 전문가 시스템에서는 실시간내에 추론을 할 수 있도록 퍼지 추론에서의 퍼지 규칙들에 대한 병렬화가 필요하다. 따라서 퍼지 규칙의 전건부와 후건부에 대해서 변수들을 각각 병렬로 추론할 수 있는 일반적인 퍼지 병렬화 추론 방법과 알고리즘 및 이에 적합한 개념으로 설계된 병렬 퍼지 아키텍처의 설계에 대한 연구가 필요하다. 따라서, 본 논문에서는 다음과 같은 요구사항을 만족하는 퍼지 정보처리에 적합한 병렬 퍼지 아키텍처를 설계하여 평가한다.

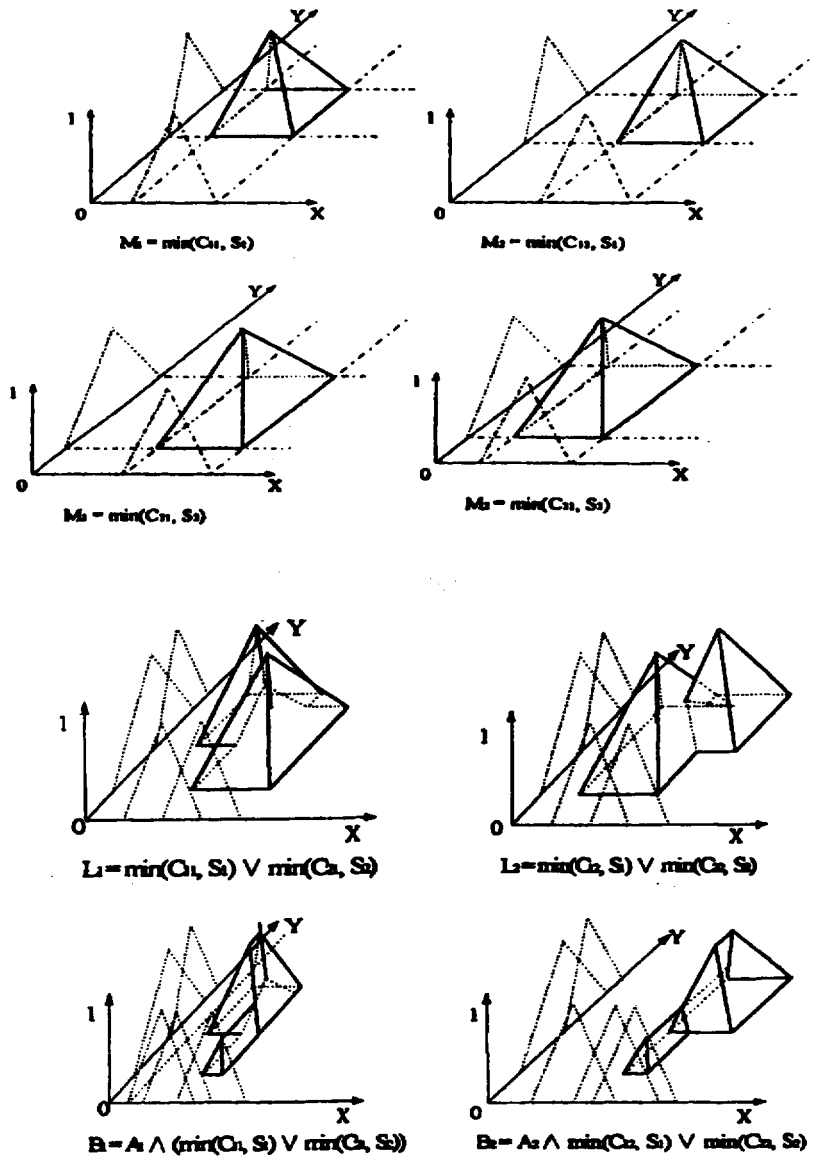
- 퍼지 추론에 있어서 효율적인 병렬 추론 방법
- 효율적인 병렬 추론 방법에 따른 병렬 퍼지 아키텍처의 설계
- 전문가 시스템과 같은 대단위의 지식 처리 시스템의 구현
- 범용성, 확장성, 고속성을 갖는 퍼지 추론 시스템의 설계
- 입, 출력 변수의 수 및 각 변수의 최대 항수에 무관한 아키텍처의 설계
- 고성능을 가지면서 저가격의 퍼지 병렬 아키텍처의 구현

2. 퍼지추론 방법

퍼지추론 과정은 m개의 입력변수와 n개의 출력변수로 구성된 규칙들로 이루어진 지식 베이스로부터 각 규칙의 전건부에서 정의된 연산(Min·Max 연산)을 수행하여 이들의 최종적합도(degree of fulfillment)로부터 최종 결론을 후건부에서 얻는다. 이때 n개의 출력은 서로 독립적으로 사용될 수 있으나, m개의 입력들은 n개의 출력을 유도하기 위해 서로 종속적이며 m개의 입력이 갖는 특성에 의해 추론 결과가 직접적인 영향을 미치게 되며 바로 이러한 추론 부분은 많은 병렬성을 내포하고 있고 이러한 병렬 알고리즘을 잘 이용하면 고속 퍼지 연산이 가능해진다. 이러한 퍼지 지식처리 시스템의 특성은 고속 및 유연성의 특성을 갖고 있어야 한다. 기존의 퍼지추론 칩 및 하드웨어의 연구들은 고속으로 동작하고 있지만 하드웨어가 고정되어 있어서 범용성 및 확장성은 떨어진다. 이들이 단지 속도에는 관계없이, 입력 및 출력 변수의 수가 증가된 환경이나 입력 및 출력의 비트 크기가 달라진 환경에서 응용되어 사용될 때는 하드웨어의 확장성이 없으므로 재설계를 하거나, 추론 속도가 현저하게 저하되는 경향이 있다.

퍼지 입력 변수의 수가 m개이고, 각 변수에 대해 k개의 linguistic value가 있다면 생성 가능한 최대 제어규칙의 수는 k^m이 된다. 따라서 이렇게 많은 수의 규칙들의 전건부와 후건부의 변수들을 순차적인 방법에 의해 추론하는 것은 상당히 많은 시간을 요하므로 고속의 퍼지 추론이 필요한 시스템이나 대규모의 퍼지 전문가 시스템에는 적합하지 않다.

퍼지추론에는 여러가지의 방법이 알려져 있지만, 일반적으로 Mamdani 방식, Larsen 방식, Tsukamoto 방식, Sugeno 방식이 잘 알려져 있다. Larsen 방식과 Tsukamoto 방식은 Mamdani 방식의 변형이라고 생각할 수 있고, Sugeno 방식은 후건부를 전건부의 선형식으로 구성한다. 본 논문에서는 Mamdani 방식 및 Koczy 방식[9,10]에 대해서, 퍼지 규칙의 전건부와 후건부에 대해서 변수들을 각각 병렬로 추론할 수 있는 일반적이고 효율적인 퍼지 병렬화 추론 방법과 알고리즘을 제안하고 이러한 병렬 퍼지추론에 적합한 고속이면서 유연성 및 확장성을 갖는 퍼지 병렬 컴퓨터를 설계하여 범용의 퍼지 컴퓨터로서 퍼지 추론 및 대규모의 퍼지 전문가 시스템에도 활용될 수 있도록 한다.



(그림 1) 알고리즘 2에 의한 퍼지추론 방법
(Fig. 1) Fuzzy inference method of Algorithm 2

알고리즘 1은 Mamdani가 제안한 Min · Max 연산을 이용하여 퍼지 합성규칙을 도입한 것으로 추론방식 중에서 가장 많이 사용되어지고 있다.

【 알고리즘 1 】

```
/* r: 규칙의 수 */
/* m: 전건부 변수의 수 */
/* n: 후건부 변수의 수 */
   For i = 1 to r
      For j = 1 to m
         dj = max (Cij(x) ∧ Aj(x))
         Di = min(Di, dj)
```

```
    For k = 1 to n
        Pik(y) = min(Di, Sik(y))
  let B(y) = 0
  For i = 1 to r
    For k = 1 to n
      Bk(y) = Bk(y) ∨ Pik(y)
  For k = 1 to n
    bk = Σ (yi ∈ Y-Yi) yi · Bk(yi) / Σ (yi ∈ Yi) yi
```

알고리즘 1은 입력값과 규칙의 전건부와의 Min·Max 연산을 행한 후에 얻어진 적합도와 후건부의 연산을 통해 비퍼지화의 값을 구한다. 이 알고리즘은 제어 규칙의 전건부와 후건부를 모두 일반언어로 대응시킬 수가 있고, 정성적으로 알기쉽다는 점과 추론과정을 그림으로 표현하기 쉽다는 장점이 있다. 알고리즘 2 [9,10]에서는 사전에 각각의 규칙의 전건부와 후건부의 Min·Max 연산을 수행한다. 그리고, 앞에서 계산된 값과 입력값의 Min 연산에 의해 입방체의 부피의 무게중심을 구하는 방법이다.

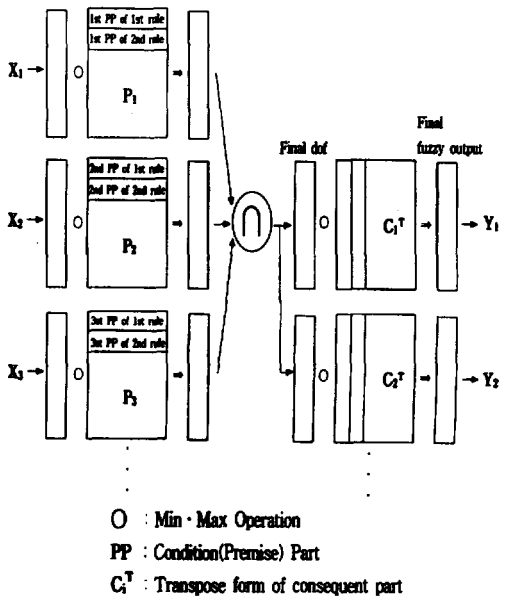
【 알고리즘 2 】

```
/* r: 규칙의 수 */
/* m: 전건부 변수의 수 */
/* n: 후건부 변수의 수 */
  For every (x, y)
      let L(x, y) = 0
      let B(x, y) = 1
  For i = 1 to r
    For j = 1 to m
      For k = 1 to n
        Djk(x, y) = min(Cij(x) ∧ Sik(y))
        Ljk(x, y) = Ljk(x, y) ∨ Djk(x, y)
  For k = 1 to n
    For j = 1 to m
      P(x, y) = Aj(x) ∧ Ljk(x, y)
      Bk(x, y) = Bk(x, y) ∧ P(x, y)
  For k = 1 to n
    bk = Σ (x ∈ X, yi ∈ Y-Yi) Σ y · Bk(x, y) / Σ (yi ∈ Yi) yi
```

알고리즘 2에서의 전건부와 후건부의 Min 연산은 알고리즘 1의 Min 연산과는 달리 어떤 특정한 값이 아니라 그림 1에 나타낸 것처럼 전건부 변수와 후건부 변수에 대해 피라미드 형태의 3차원 모양의 값이 나오게 된다. 따라서 무게중심을 구하는 단계에서도 알고리즘 1과는 달리 부피에 대한 무게중심의 값을 계산해 주어야 한다.

3. 퍼지 규칙의 병렬추론 방법

퍼지추론을 이용하여 복잡한 대상에 대한 제어 시스템을 구성하는 경우, 전건부와 후건부의 수가 증가함에 따라 추론시간도 증가하게되므로 퍼지추론의 병렬화 방법이 요구된다. 따라서, 알고리즘 1, 알고리즘 2에 대해, 퍼지규칙들을 단위로 하는 새로운 병렬 퍼지추론 방법이 필요하게 된다.



(그림 2) 퍼지 병렬 추론 방법 1
(Fig. 2) Parallel fuzzy inference method 1

알고리즘 1에 대한 병렬화 방법으로서는 전건부와 후건부의 변수들을 독립시켜, 각각의 규칙에 대해 전건부 및 후건부 변수들을 병렬로 추론하는 방법으로 그림 2에 나타나 있다.[6] 여기서 연산자 ○는 Min·Max의 연산을 나타내고, X_i는 i번째 입력변수의 입력 벡터값, P_i는 i번째 전건부의 소속함수 행렬이다. 퍼지화된

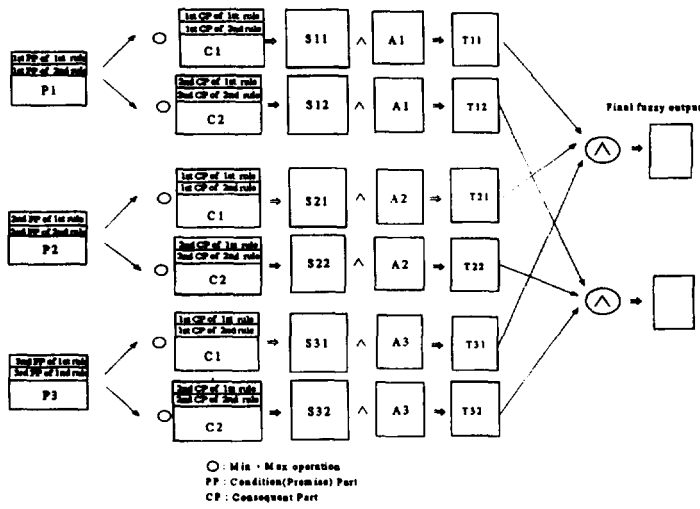
센싱 입력 X_1과 P_1의 Min · Max 연산을 하면 첫번째 전건부의 적합도가 나온다. 마찬가지로 X_2와 P_2를 Min · Max 연산하면 두번째 전건부의 적합도가 나온다. 따라서 출력 행렬은 규칙의 수만큼의 항을 가지게 된다. 이러한 방법으로 병렬로 연산된 r(규칙의 수)개의 적합도의 Min 연산을 취하면 전체 규칙에 대한 최종적 합도가 나오게 된다. 이것을 후건부의 변수의 개수만큼 broad- cast하여, 전건부의 각 변수와 같은 방법으로 Min · Max 연산을 취하면, 최종적인 퍼지 출력이 후건부의 변수의 개수만큼 각각 계산되어 진다. 이러한 병렬 퍼지 컴퓨터의 동작 순서는 다음과 같다.

【 알고리즘 3 】

1. 센싱된 퍼지화 입력을 입력 레지스터 X_i에 저장한다. (병렬)
2. X_i와 P_i간의 Min · Max 연산을 병렬로 수행하여 각각의 전건부 변수에 해당하는 적합도를 계산한다. (병렬)
3. 각각의 PE_i에서 연산된 적합도를 shifter로 옮긴다. (병렬)
4. 각각의 적합도의 Min연산을 위해 Min network를 통해 bit-wise로 규칙의 수만큼의 minimum 값을 구한다. (순차처리)
5. 앞에서의 minimum 값이 다 구해지면 CP의 min_store_register에 저장한다.
6. min_store_register에 저장된 값을 bit-wise로 Broadcast network를 통해 각각의 PE_i에 broadcast한다. (병렬)
7. broadcast된 전체규칙의 최종적합도와 C_i^T(후건부 변수의 전치행렬)를 Min · Max 연산하여 최종출력을 구한다. (병렬)

위에서 제안한 병렬화된 구조에서는 전건부의 최종 적합도를 구하는 곳에서만 순차적 방식으로 진행이 되고 그 외의 부분에서는 모두 병렬로 수행된다.

알고리즘 2에 대한 병렬연산은 알고리즘 1을 위한 것보다 더 복잡한 작업이 이루어진다. 알고리즘 1에서는 전체 규칙의 전건부와 입력값과의 연산을 모두 수행하고 나온 적합도와 후건부의 값을 연산하는 방식이지만, 알고리즘 2는 미리 각 규칙의 전건부와 후건부와의 연산을 통해 값을 구해야 한다. 따라서 전건부의 퍼지 변수의 개수가 m개이고 후건부의 퍼지 변수가 n개 일 경우 전건부와 후건부 사이에 m × n개의 연산이 필요하게 된다. 가령 전건부의 퍼지 변수의 수가 3개이고 후건부의 퍼지 변수의 수가 2개인 시스템에서는, P_1에는 각 규칙의 전건부의 첫번째 퍼지 변수를 저장하고, P_2에는 각 규칙의 전건부의 두번째 퍼지 변수를, P_3에는 세번째 퍼지 변수로 저장되고, C_1에는 각 규칙의 첫번째 후건부 값이 C_2에는 두번째 후건부의 값이 저장이 된다. 이때 각각 P_1, P_2, P_3와 후건부 C_1, C_2사이에 Min · Max연산을 취하여 그 값을 저장한다. 즉 Min ·



(그림 3) 병렬 퍼지 추론 방법 2
(Fig. 3) Parallel fuzzy inference method 2

$Max(P_1, C_1)$, $Min \cdot Max(P_1, C_2)$, $Min \cdot Max(P_2, C_1)$, $Min \cdot Max(P_2, C_2)$, $Min \cdot Max(P_3, C_1)$, $Min \cdot Max(P_3, C_2)$과 같은 3 × 2인 6번의 연산을 통해 S값이 계산되어 진다. 이 연산된 값들과 입력값 A와의 Min연산을 $Min(S_{11}, A_1)$, $Min(S_{12}, A_1)$, $Min(S_{21}, A_2)$, $Min(S_{22}, A_2)$, $Min(S_{31}, A_3)$, $Min(S_{32}, A_3)$과 같이 수행한후 T에 저장한다. 그리고 $(T_{11} \wedge T_{21} \wedge T_{31})$, $(T_{12} \wedge T_{22} \wedge T_{32})$를 통해 후건부 n개에 대한 결과값을 얻는다. 알고리즘 2의 병렬화 방식은 그림 3에 나타나 있고, 연산의 순서는 알고리즘 4와 같다.

【 알고리즘 4 】

1. 입력 레지스터 A_i에 퍼지화된 입력값을 넣는다. (병렬)
2. P_i와 C_i^T사이의 Min · Max 연산을 병렬로 수행하다. (병렬)
3. P_i와 C_i^T의 Min · Max 연산을 통해 계산된 값을 레지스터 S에 저장한다. (병렬)
4. S_{ij}와 A_i의 값의 Min연산을 병렬로 수행한다. (병렬)
5. Min 연산을 통해 계산된 값을 레지스터 T_{ij}에 저장한다. (병렬)
6. T_{ij} 값들의 Min 연산을 취하여 결과값을 얻는다. (순차처리)

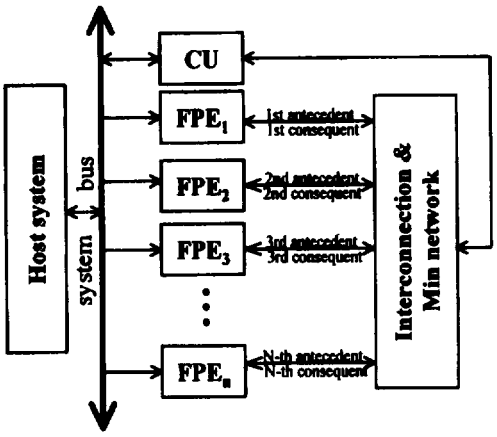
이와 같은 알고리즘 4에서는 알고리즘 3과 같이 전건부와 후건부의 변수들을 독립시켜, 각각의 규칙에 대해 전건부 및 후건부의 변수를 병렬로 추론하게 된다. 그러나, 알고리즘 4에서는 입력값과는 별도로 각 규칙들과 각 규칙의 전건부, 후건부를 통해 미리 계산된 값 S를 가지게 된다. 따라서 시스템의 초기에 일단 S에 대한 값의 연산이 이루어 진 후에는 이미 계산된 S와 입력값 A와의 연산만을 통해 결과값을 얻을 수 있으므로 상황에 따라서는 더 효과적이라고 볼 수 있다. 따라서 학습 능력에 있어서 규칙의 소속함수의 값이 계속적으로 변화하는 시스템에서는 알고리즘 3이 우수하지만 고정된 소속함수의 값을 통해 제어를 하는 분야에서는 알고리즘 4가 더 우수하다. 또한 알고리즘 4의 복잡한 연산을 병렬로 수행함으로써 계산량을 감소시킬 수 있다.

4. 퍼지 병렬 아키텍처

과거 수년동안 고속 퍼지추론을 위한 여러 가지의 특수한 하드웨어가 개발되었지만 이들 대부분은 정해진 추론 방법만을 위해서 동작하고 있으며, 여러 가지 형태의 퍼지집합 연산이나 퍼지규칙들을 위한 병렬추론 방법은 지원하고 있지 않다. [7,8]에서 KAFA(KAIST Fuzzy Accelarator)라고 하는 퍼지 하드웨어 아키텍처를 개발하였는데 이것은 여러 가지의 퍼지추론 방법 및 퍼지집합 연산을 지원한다. 이 아키텍처는 크게 시스템 제어장치(CU)와 산술연산장치인 FPE(fuzzy processing element)의 두 부분으로 구성되어 있는데 FPGA (Field Programmable Gate Array) 칩들을 사용하여 구현하였다. 각각의 FPE는 최고속도 5MFLOPS를 갖는데, KAFA에는 128개의 FPE가 있으므로 최고성능은 640 MFLOPS에 달한다. 128개의 FPE로 구성된 배열은 SIMD 모드로 동작하고 있다. 그러나, KAFA에서는 각각의 FPE는 주어진 소속함수의 하나의 원소만의 명령을 수행한다. 실제로 많은 경우에 하나의 퍼지집합의 소속함수는 좁은 범위의 형태를 갖고 있으므로, 많은 수의 FPE는 사실상 필요하지 않게 된다. 대부분의 경우, 많은 FPE는 비효율적이다. 따라서 KAFA의 방식은 전건부, 후건부 또는 퍼지규칙들을 병렬화하는 방법이 아니고, 하나의 소속함수들의 처리에 있어 이들의 양자화된 원소들을 병렬처리하는 일종의 fine-grain parallelism 방식이다. 이러한 형태는 사실상 디지털 게이트 또는 퍼지전용 칩들로부터 쉽게 구현할 수 있다.

본 논문에서는 KAFA에서의 연산처리의 비효율성을 극복하기 위하여 다입력 · 다출력 시스템에 적합한, 확장가능한 퍼지병렬 아키텍처를 제안한다. 그림 4에 앞에서 논한 두가지의 병렬처리 방식에 적합한 새로운 병렬 퍼지추론의 모델을 나타낸다. 여기에는 n개의 FPE가 있고, 각각의 FPE_i는 i번째의 전건부와 i번째의 후건부의 연산만을 수행한다. 전건부에서의 모든 입력변수들의 처리는 병렬로 수행되어지고, 후건부에서도 모든 출력변수들은 병렬로 수행되어진다. 그러므로 각각의 FPE들은 최대로 활용(fully utilize)되어진다. 필요한 FPE의 수는 전건부와 후건부의 변수의 최대수와 같다. 예를들어 전건부의 변수가 5개, 후건부의 변수가 3개 있으면 필요한 FPE의 수는 5가 된다. 이러한 시스템은 일반적이고 확장가능한 모듈형의 퍼지병렬 아키

텍처이다. 이와 같은 구성은 많은 입출력 변수들을 갖는 경우에 매우 효율적이다.

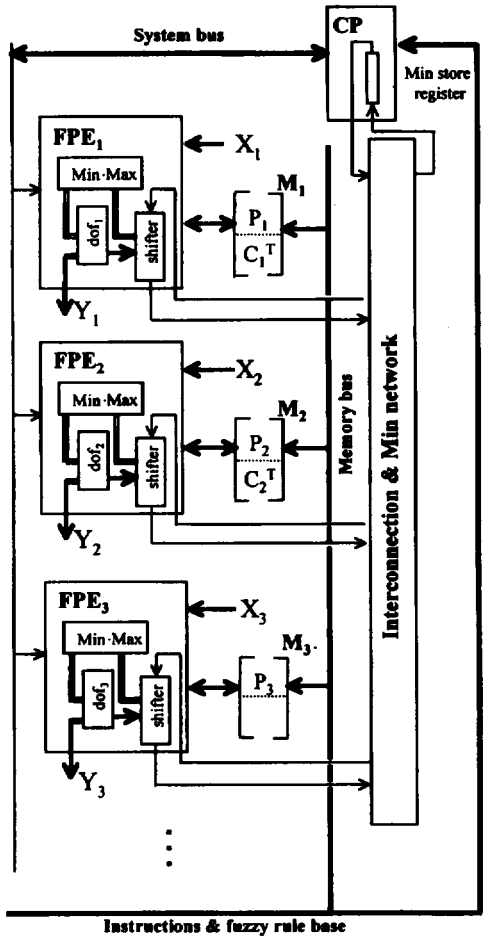


(그림 4) 병렬 퍼지 추론 시스템의 모델
(Fig. 4) Model of parallel fuzzy inference system

그림 5에 n개의 FPE를 갖는 상기의 구조를 위한 SIMD 형태의 구조를 나타낸다. 이것은 여러개의 FPE, 하나의 CP, 그리고 메모리 모듈들로 구성되어 있다. 각각의 FPE는 자신의 고유 메모리 모듈과 연결되어 있다. 퍼지규칙들과 데이터들은 메모리 모듈들을 통하여 분산되어 있다. 각각의 FPE는 퍼지추론을 위한 Min · Max 연산, 적합도 연산 및 쉬프트 연산을 수행한다. CP는 호스트 시스템과 FPE들의 연산을 제어하는 인터페이스로서의 역할을 한다. 이러한 아키텍처는 성능의 손실 없이 모델의 크기, 입출력 변수들을 증가시킬 수 있다. FPE들은 SIMD 구조의 형태로 연결된 시스템으로, 이러한 구조를 택한 이유는 퍼지 추론에서 규칙의 모양이 거의 정형적이며, 소수의 연산이 집중적으로 사용되기 때문이다. 그리고 입력 데이터들을 메모리에 저장하여 필요한 경우에 메모리를 참조하는 것보다는 레지스터 파일에 미리 적재 시킨 후에 필요할 때에 레지스터 참조 명령으로 수행하는 것이 훨씬 효율적이다. 또한 레지스터간의 연산에 의한 추론에서는 Min · Max 연산이 대부분인데 이 경우에는 SIMD 구조를 사용하였으므로 추론 속도는 상당히 높아진다.

위와 같은 구조를 기본으로 제안된 아키텍처를 다음과 같은 파라메터들을 사용하여 설계하였다.

- 전건부의 변수의 수 : 5
- 후건부의 변수의 수 : 3
- antecedent의 level 수 : 7
- consequent의 level 수 : 7
- universe of discourse sample : 256
- 전체규칙 수 : 512
- 소속함수 encoding에 필요한 bit 수 : 6



(그림 5) 퍼지 병렬 아키텍처
(Fig. 5) Parallel fuzzy architecture

표 1은 퍼지추론을 각종의 개발 장치로 수행한 경우의 추론 속도와의 비교를 나타낸다. 비교는 두가지 경우로 나누어 각각 벤치마크 테스트하였다. 즉, 2입력, 1출력 변수일때의 규칙의 수가 7개인 시스템과 3입력, 2출력 변수일 때의 규칙의 수가 14개인 시스템으로 설

정하였다. 모든 경우에 대해 각 입출력 변수의 linguistic value의 수는 3개 (SM, ME, LA)인 간단한 시스템으로 하였다. 비교대상으로는 8-bit 마이크로컨트롤러 8051, PC 486, 퍼지 전용칩인 Togai FC110으로 하였다. FC110은 일반 마이크로프로세서나 마이크로컨트롤러보다 성능이 우수하지만, 제안된 시스템보다는 성능이 떨어짐을 알 수 있다. 또한, FC110은 입출력의 수, 퍼지규칙의 수 및 추론방법들이 제한되어 있으므로 확장성이나 유연성은 없다. 표 1에서 알 수 있듯이 제안된 시스템은 2입력-1출력의 경우보다 3입력-2출력의 경우가 더욱 더 높은 speedup을 나타내고 있다. 예를들어 FC110과 비교할 경우, 2입력-1출력에서는 10.5배이고, 3입력-2출력의 경우에는 12.0배로 나타났다. 본 논문에서 제안한 시스템은 전건부 및 후건부의 변수들을 병렬로 처리하므로, 입출력 변수들이 증가되면 speedup은 더욱 증가된다.

〈표 1〉 성능 비교
〈Table 1〉 Performance comparison

	8051	i486 (33MHz)	FC110	Proposed
2입력, 1출력, 7규칙	2860	83	42	4
3입력, 2출력, 14규칙	8730	225	84	7

(계산 시간 단위는 μs)

본 논문에서 제안한 퍼지병렬 추론 아키텍처의 주요한 특징을 요약하면

- 고속 (120 MFLOPS)
- 전건부, 후건부의 변수의 수에 관계없이 쉽게 구성가능
- 설계하기에 용이하며 VLSI화를 가능하게 하는 디지털 설계
- 복잡한 퍼지규칙도 사용가능
- 성능의 손실없이 모델크기 증가 가능

등의 특성을 얻을 수 있다.

5. 결 론

본 논문에서는 퍼지규칙들에 대해 전건부 및 후건부들을 각각 병렬로 추론할 수 있는 새로운 병렬 퍼지추론 방법을 제안하고, 대규모 전문가 시스템 등에서의 퍼지 정보처리에 적합한 효율적인 아키텍처를 설계 및 구현하였다. 제안된 퍼지병렬 아키텍처는 전건부 변수들의 최종 적합도값을 구하는 부분에만 순차적으로 동작하고, 나머지 부분들은 전부 병렬로 동작하므로, FC110과 같은 퍼지 전용칩보다 10배 이상의 속도로 처리된다. 따라서 상당히 높은 성능을 나타내고, 입출력 변수 및 규칙베이스에 무관한 일반화된 구조로 되어 있다. 또한, 이 시스템에서 각각의 FPE의 효율은 상당히 높다. 특히 이러한 시스템은 MISO (Multiple-input single-output) 시스템보다는 MIMO (Multiple-input multiple-output) 시스템에 더욱 효율적이다. 이러한 병렬 퍼지추론 아키텍처는 실시간 시스템과 같이 고속의 추론시간을 요하는 시스템 및 전건부와 후건부에 많은 수의 추론변수들을 갖고 있는 대규모 전문가 시스템이나 영상인식 시스템, 이동 물체의 인식, 의료진단 시스템, 기타 고속 퍼지 추론이 요구되는 시스템 등에 사용될 수 있다. 향후의 연구로서는 전문가 시스템과 같은 실제의 MIMO 시스템에 제안된 아키텍처를 적용하여 성능평가를 할 필요가 있다.

참 고 문 헌

[1] G. Ascia and et al., "Designing for Parallel Fuzzy Computing," *IEEE Micro*, pp.62, Dec. 1995.

[2] T. C. Chiueh, "Optimization of Fuzzy Logic Inference Architecture," *Computer*, pp.67-70, May 1992.

[3] H. Eichfeld and et al., "A General-Purpose Fuzzy Inference Processor," *IEEE Micro*, pp.12-17, Jun. 1995.

[4] A. Jaramillo-Botero, "Parallel, High-speed PC Fuzzy Control," *IEEE Micro*, pp.63, Dec. 1995.

[5] A. Kandel, "Fuzzy Hardware Challenges," *IEEE Micro*, pp. 61, Dec. 1995.

[6] H. Kang and G. Vachtsevanos, "Fuzzy Hypercube: Linguistic Learning/Reasoning Systems for Intelligent Control and Identification," *Journal of Intelligent and Robotic Systems*, Vol.7, pp.215-232, 1993.

[7] Y. D. Kim and et al., "Parallel Fuzzy

Information Processing System," *Fuzzy Sets and Systems* 72, pp.323-329, 1995.

[8] Y. D. Kim and H. Lee-Kwang, "High Speed Flexible Fuzzy Hardware for Fuzzy Information Processing," *IEEE Tr. on SMC*-Part A, Vol.27, No.1, pp.45-56, Jan. 1997.

[9] L. T. Koczy and K. Hirota, "A Fast Algorithm for Fuzzy Inference by Compact Rules," *Fuzzy logic for the management of uncertainty*, pp.297-317, 1992.

[10] L. T. Koczy and K. Hirota, "Fuzzy Inference by Compact Rules," *Proc. of Int. Conf. Fuzzy Logic & Neural Network*, IIZUKA'90, Iizuka, pp.307-310, 1990.

[11] H. Li and M. Gupta, *Fuzzy Logic and Intelligent Systems*, Kluwer Academic Publishers, Boston, 1995.

[12] L. Salvador and et al., "A Multilevel Systolic Approach for Fuzzy Inference Hardware," *IEEE Micro*, pp.61-71, Oct. 1995.

[13] H. Surmann and et al., "Fuzzy Rule-Based System on General-Purpose Processors," *IEEE Micro*, pp.40-48, Aug. 1995.

[14] H. Surmann and et al., "What Kind of Hardware is Necessary for a Fuzzy Rule Based System," *Proc. FUZZ-IEEE WCCI*, Vol.26, No.6, pp.274-278, 1994.

[15] M. Togai and H. Watanabe, "Expert System on a Chip: An Engine for Real-time Approximate reasoning," *IEEE Expert Syst. Mag.*, Vol.1, pp.55-62, 1986.

[16] A. P. Ungering and K. Goser, "Architecture of a 64-bit Fuzzy Inference Processor," *Proc. FUZZ-IEEE WCCI*, Vol.26, No.6, pp. 1776-1780, 1994.

[17] H. Watanabe and et al., "A VLSI fuzzy Logic Controller with Reconfigurable, Cascadable Architecture," *IEEE Journal of Solid State Circuits*, Vol.25, No.2, pp.376-382, Apr. 1990.

[18] H. Watanabe, "RISC Approach to Design of Fuzzy Processor architecture," *FUZZ-IEEE 92*, pp.431-441, 1992.

[19] T. Yamakawa, "Intrinsic Fuzzy Electronic Circuits for Sixth Generation Computer," in *Fuzzy Computing*, M. M. Gupta and M. Yamakawa (eds.), North-Holland, pp.157-171, 1988.

[20] T. Yamakawa, "Silicon Implementation for a Novel High-speed Fuzzy Inference Engine: Mega-FLIPS Analaog Fuzzy processor," *Journal of Intell. and Fuzzy Systems*, Vol. 1, No.1, 1993.

이 상 구

1978년 서울대학교 전자공학과 (공학사)

1981년 한국과학기술원 전산학과 (석사)

1995년 한국과학기술원 전산학과 박사수료

1978년~1979년 서울대학교 전자공학과 조교

1981년~1983년 국방과학연구소 연구원

1988년~1989년 동경대학 정보공학과 연수

1983년~현재 한남대학교 컴퓨터공학과 교수

관심분야 : 컴퓨터 구조(마이크로 프로세서, 퍼지 컴퓨터)병렬처리