

□ 特 輯 □

# 퍼지 컴퓨터

한국과학기술원 전산학과 이 광 형\*  
 한국과학기술원 전기전자과 박 규 호\*

## 목 차

1. 서 론	3. 퍼지 논리 회로
2. 퍼지 단어의 자료구조와 퍼지추론	4. 퍼지 컴퓨터의 구조
	5. 결 론

### 1. 서 론

이진논리에 기초한 기존의 디지털 컴퓨터는 정확한 값들만을 다루기 때문에 우리가 일상생활에서 흔히 접하게 되는 애매한 정보를 처리하기에는 부적합한 면이 있다. 1965년 자테교수에 의해 퍼지 이론이 소개된 이후 애매함을 처리하는 많은 분야에서 퍼지 이론이 응용되었으며, 이에 관련된 연구가 지금도 활발히 진행 중에 있다. 퍼지 이론의 응용은 이진 논리에 기초를 두고 있지 않음에도 불구하고, 많은 경우 범용의 디지털 컴퓨터에서 개발 실행되어 왔다. 비록 범용의 디지털 컴퓨터에서 퍼지 논리를 시뮬레이션을 할 수는 있었지만 이것은 너무 비효율적이고 퍼지논리를 제대로 수행한다고 볼 수 없었다. 이것이 퍼지 논리에 적합한 퍼지 컴퓨터에 관한 연구의 동기가 되었다.

퍼지 컴퓨터에 관한 연구는 80년대 중반부터 몇 가지 모델이 발표되면서 시작되었다. 크게 두 가지 방향으로 연구가 진행되었는데 하나는 기존의 디지털 회로를 이용하여 퍼지 집합을 표현하고 이를 빨리 처리할 수 있

도록 하는 추론기관을 VLSI화하는 방향이고[11], 또 하나는 소속함수 값을 표현하는 0과 1 사이의 값을 아날로그 회로의 0V에서 5V 사이의 값으로 대응시켜 처리하는 아날로그형 컴퓨터[12]이다. 물론 후자의 경우가 처리속도가 빠르고 기억장소를 적게 사용하지만, 디지털 시스템의 견고성과 프로그래밍의 용이성 때문에 전자의 방식을 사용한 퍼지 프로세서가 상품화되어 사용되고 있다.

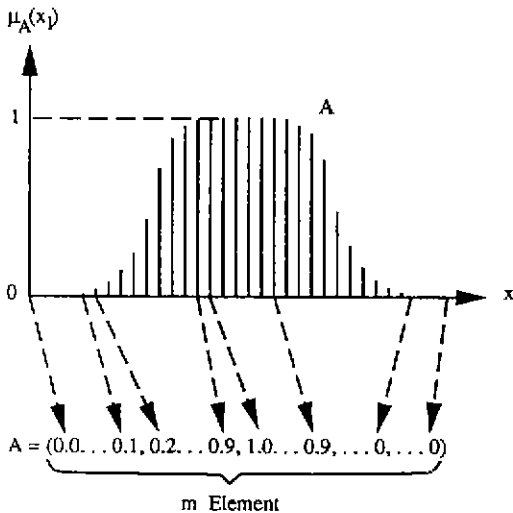
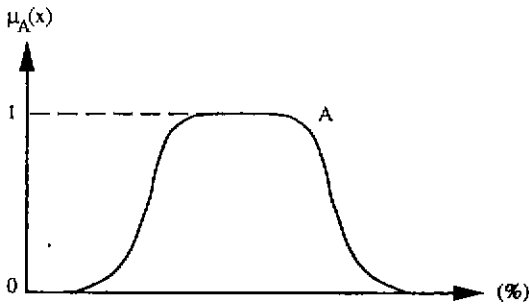
퍼지 컴퓨터에 관한 연구는 아직 제한적인 범위 내에서 이루어지고 있다. 이제까지 나온 퍼지 컴퓨터는 엄밀히 말해서 퍼지 논리를 기초로 한 퍼지 추론기관이라 할 수 있다. 또한 연구 결과에 대한 체계적인 정리가 되어 있지 않았으며, 제안된 컴퓨터 구조 사이에 장단점을 비교할 수 있는 단계에 이르지 않은 상태이다. 여기에서는 지금까지 제안된 퍼지 컴퓨터 연구 중에서, 특히 하드웨어에 중점을 두어 몇 가지 연구 결과를 소개한다. 먼저 퍼지 컴퓨터가 다루는 퍼지 단어의 자료구조를 살펴보고, 퍼지추론에 대해 자세히 알아본다.

### 2. 퍼지 단어의 자료구조와 퍼지추론

\*중심회원

2.1 퍼지 단어의 자료구조

일반적으로 우리는 일상생활 속의 여러가지 경험을 통해 언어 정보 형태로 지식을 획득한다. 이러한 언어 정보는 애매한 형태로 이루어질 때가 많다. 예를 들어, "환자의 체온이 며칠동안 높으면 감기에 걸렸을 것이다"에서 '높다'는 정보는 상황에 따라 여러가지로 해석되는 애매한 단어이다. 이러한 애매한 단어들은 퍼지집합으로 표현될 수 있다. 퍼지집합은 소속함수 값을 가지는데 일반적으로 이 함수의 모양은 다양한 형태의 연속함수이다. 사실 컴퓨터에서 다양한 연속함수를 표현한다는 것은 상당한 어려움이 따른다. 따라서 많은 경우 전체 집합 내에서 m개를 샘플링하여 각각을 원소로 가지는 벡터로 표현한다. 즉, 퍼지집합은 (그림 1)에서와 같이 m개의 원소를 가지는 벡터로 표현할 수 있다.



(그림 1) m원소 퍼지집합에 의한 퍼지단어 표현

원소의 소속정도를 기존의 디지털 시스템으로 표현할 때 각 원소에 대해 할당하는 비트수가 n개라면 소속정

도는  $2^n$ 등급으로 구분되어 진다. 따라서 m개의 원소를 가진 하나의 퍼지집합을 표현하기 위해서는  $m \cdot n$  bit의 기억장소가 필요하다. 비록 이러한 방식으로 퍼지집합을 처리하기 위해서는 많은 기억장소의 시간이 필요하지만 프로그래밍이 편리하다는 이점이 있다.

원소의 소속함수를 표현하는 다른 방법으로 0V와 5V 사이의 전압값으로 표현하는 아날로그 회로가 있다. 하나의 퍼지 단어는 m개의 신호선으로 표현되며, 각 신호선은 소속정도에 따른 전압을 가지게 된다.

2.2 퍼지 추론

퍼지추론을 하는 가장 기본적인 규칙은 generalized modus ponens과 generalized modus tollens이다. A, B, A', B'를 퍼지 언어변수의 값(퍼지집합)들이라 할 때 퍼지추론은 다음과 같이 표현된다.

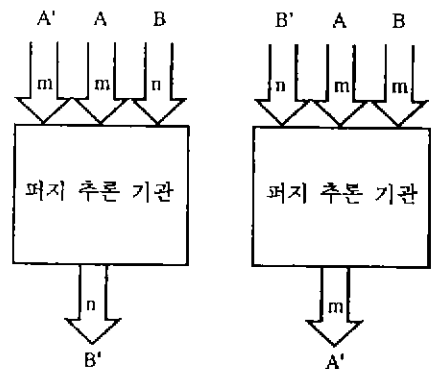
① generalized modus ponens(전방향 추론 : forward inference)

$$\frac{\begin{array}{l} \text{If } X \text{ is } A, \text{ then } y \text{ is } B \\ X \text{ is } A' \end{array}}{y \text{ is } B'}$$

② generalized modus tollens(후방향 추론 : backward inference)

$$\frac{\begin{array}{l} \text{If } X \text{ is } A, \text{ then } y \text{ is } B \\ y \text{ is } B' \end{array}}{X \text{ is } A'}$$

Generalized modus ponens는 주로 제어(control)에 많이 이용되고, generalized modus tollens는 의학 진단 등의 전문가 시스템에서 이용된다.



(a) 전방향 추론 (b) 후방향 추론

(그림 2) 퍼지 추론기관

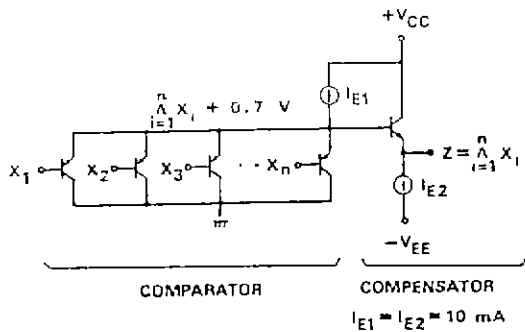
이러한 추론에서는 퍼지집합 A와 A'(또는 B와 B')가 어느 정도 일치하는가 하는 것을 판정하는 일이 매우 중요하다. 그리고 결론의 B중에서 일치하는 정도만큼의 부분을 B'로 출력한다. 이러한 추론을 하기 위해서 (그림 2)와 같은 퍼지 추론기관(fuzzy inference engine: FIE)을 생각할 수 있다.

이 추론기관에서 일어나는 기본적인 연산은 MIN, MAX 연산이다. A'의 입력에 대해 A의 일치 정도를 알아보기 위해 두 퍼지집합을 MIN하고 이렇게 나온 퍼지집합에서 MAX인 값을 B에 적용하여 B'를 구하는 등 연속된 MIN, MAX 연산으로 추론이 이루어진다.

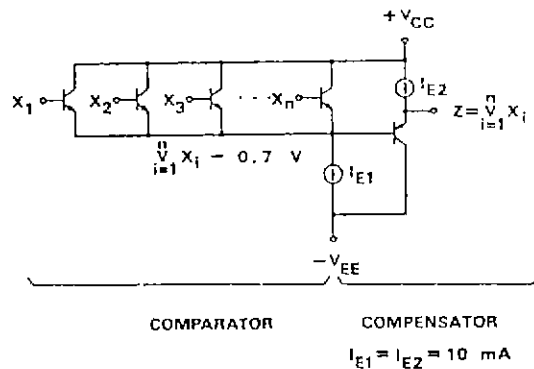
### 3. 퍼지 논리 회로

#### 3.1 MAX, MIN 회로

트랜지스터를 이용하여 만든 퍼지추론에 가장 기본이



(그림 3) MIN 논리회로



(그림 4) MAX 논리회로

되는 Max와 Min 연산을 위한 회로가 (그림 3)과 (그림 4)에 있다. 여기에서 Min 연산 회로에서는 PNP형 트랜지스터를 사용하였고, Max 회로에서는 NPN형 트랜지스터를 사용하였다. 이 회로의 입출력 특성을 살펴보면, 예를 들어  $X_2=X_3=\dots=1V$ 라 할 때  $X_1$ 이 0부터 증가할 때 1까지는 Y값이 선형으로 증가하고,  $X_1=1$  후에는 출력이  $Y=1$ 로 일정하다. 또,  $X_2=X_3=\dots=2$ 이면,  $X_1=2$ 까지는 Y값이 선형 증가하고 그 후에는  $Y=2$ 로 일정하다.

#### 3.2 퍼지 플립플롭(fuzzy flip flop)

J-K 플립플롭은 1비트의 정보를 기억할 수 있기 때문에 기억장치에 많이 이용되고 있다. 이 플립플롭은 현재 상태 Q(t)와 입력(set) J(t)와 입력(reset) K(t)에 의해서 출력(다음 상태) Q(t+1)을 생성하는 함수로 특징지어진다. 일반적으로 J(t), K(t), Q(t)를 간단히 각각 J, K, Q라 쓰기로 한다 J-K 플립플롭의 특성을 minterm과 maxterm로 표시하면 아래와 같다.

$$Q(t+1) = JQ + K\bar{Q} \quad (1)$$

$$Q(t+1) = (J+Q) \cdot (\bar{K} + \bar{Q}) \quad (2)$$

여기서 식 (1)과 (2)는 드 모르강(De Morgan)의 법칙, 교환법칙, 분배법칙 등을 적용하여 동일함을 증명할 수 있다. 그러나 퍼지논리에서는 complemented law가 유효하지 않기 때문에 위와 같이 두 식의 동일함을 증명할 수 없다. 위의 식을 퍼지화하면 아래와 같다.

$$Q_R(t+1) = \{J \wedge (1-Q)\} \vee \{(1-K) \wedge Q\} \quad (3)$$

$$Q_S(t+1) = \{J \vee Q\} \wedge \{(1-K) \vee (1-Q)\} \quad (4)$$

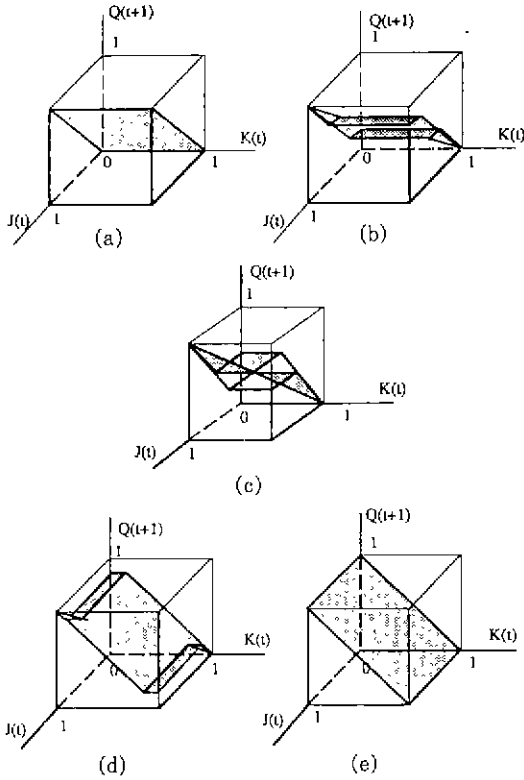
이 두 식은 서로 다른 값을 가지는데  $Q_R(t+1)$ 을 Min Max reset형 퍼지 플립플롭,  $Q_S(t+1)$ 을 Min Max set형 퍼지 플립플롭이라 부른다. 두 식을 합하기 위하여 두 식의 값이  $J=K$  직선상에서 연속적으로 연결됨을 이용하여 아래의 식을 도출해 낼 수 있다.

$$Q(t+1) = \{J \vee (1-K)\} \wedge \{J \vee Q\} \wedge \{(1-K) \vee (1-Q)\} \quad (5)$$

이 식은  $J \geq K$ 일 때는  $Q_S$ 를,  $J < K$ 일 때는  $Q_R$ 를 취했음을 알 수 있다.  $Q_S$ 와  $Q_R$ 은  $J=K$ 일 때 일치하기 때문에  $Q(t+1)$ 은 연속적이다. 따라서 이 식 (5)는 Min Max형 퍼지 플립플롭의 기본 함수가 될 수 있다. (그림 5)는 이 플립플롭의 특성을 나타내고 있는데, 각각

$$Q = 0, 0.25, 0.5, 0.75, 1.0$$

일 때,  $Q(t+1)$ 의 특성을 도식으로 보여주고 있다.



(그림 5) 퍼지 플립플롭의 특성  $Q(t+1)$

4. 퍼지 컴퓨터의 구조

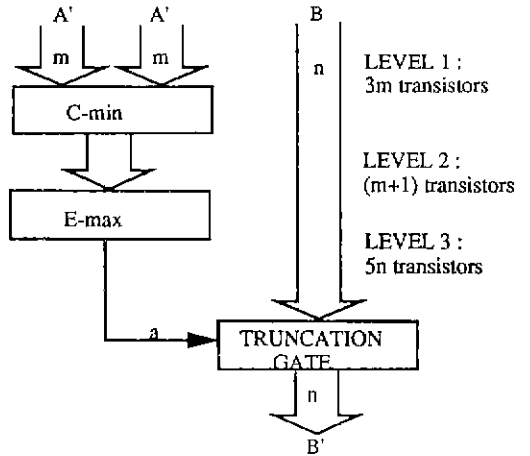
퍼지 컴퓨터는 퍼지추론을 할 수 있는 하드웨어 구조를 가져야 한다. 이 때 추론 방법은 generalized modulus ponens나 generalized modulus tollens의 근사 추론이 가능하여야 한다. 하나의 퍼지규칙(rule)을 추론할 수 있는 퍼지 추론기관(fuzzy inference engine: FIE)의 구조는 (그림 6)처럼 크게 세 개의 블럭(block)으로 되어 있다.

C-Min : Min 연산을 하는 블럭으로서, 입력으로 퍼지집합(단어) A와 A'를 받는다. 이 퍼지집합 A와 A'는 앞에서 설명한 바와 같이 m원소 퍼지집합이다(실제 구현할 때는 m=25로 한다). 이 블럭은 A와 A'의 교집합을 구하여 출력한다. 이 출력 역시 m원소의 퍼지집합이 된다.

E-MAX : 이 블럭은 Max 연산을 한다. 앞 C-MIN 블럭에서 출력한 퍼지 교집합에서 가장 높은 소속함수 값 a를 결정하여 출력한다. 즉 이 교집합의 높이

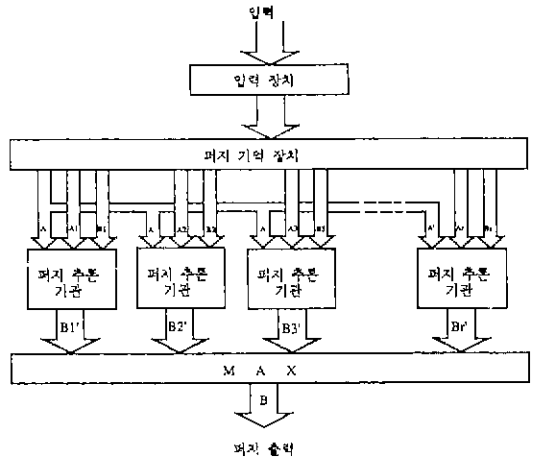
(height)가 a로 된다.

절단(truncation) : 이 절단 게이트(gate)는 n원소 퍼지 집합 B와 스칼라 값 a를 입력으로 받는다. 그리고 퍼지규칙에 저장되어 있는 퍼지집합 B를 소속정도 a 높이로 절단하여, 소속정도 a 이하를 갖는 퍼지집합 B'를 출력한다. 이때 B'도 25원소 퍼지집합이다.



(그림 6) 퍼지 추론기관의 구조(FIE)

동시에 두 개의 퍼지규칙을 적용하여 추론할 수 있도록 이 퍼지 추론기관(FIE)을 병렬로 r개 연결하면 (그림 7)과 같은 구조를 얻을 수 있다. 이와 같이 병렬로 연결하면 r개의 추론을 동시에 실행할 수 있다. 여기에서는 각각의 FIE별로 추론을 한다. 예를 들어서 세 번째 FIE는 퍼지집합 A', A<sub>3</sub>, B<sub>3</sub>를 입력으로 받아 B<sub>3</sub>'를 출



(그림 7) 퍼지 컴퓨터의 구조

력한다. 이때 모든 퍼지집합은 25 원소 집합이다.

이와 같이  $r$ 개의 추론기관이 추론한 결과  $B_1', B_2', \dots, B_r'$ 를 종합하여(합집합 연산, Max) 하나의 결과  $B'$ 를 얻기 위하여 Max 연산 회로가 있다. Yamakawa는 이러한 구조를 갖는 퍼지 컴퓨터를 실제 구현하여 1987년 IFSA(International Fuzzy Systems Association) 학술대회에 발표하였다(그림 7).

## 5. 결 론

퍼지 추론을 할 수 있는 퍼지 컴퓨터에 대해 몇 가지 사례를 중심으로 살펴보았다. 퍼지 컴퓨터를 구성하는데 아날로그 방식을 택하는 것은 퍼지 논리가 이진 논리에 기초하지 않고 0과 1 사이의 값을 가진다는 점에서 당연한 접근일 것이다. 하지만 아직까지는 퍼지 추론 외에 다른 응용분야—퍼지 데이터베이스, 퍼지 의사결정 등—에 퍼지 컴퓨터를 어떻게 접근시켜 사용할 것인가에 대해서는 정리가 되어 있지 않다. 좀더 일반적인 퍼지 컴퓨터를 정의하고 개발하는 것은 앞으로의 과제이다.

## 참 고 문 헌

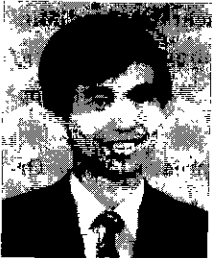
- Bortolan, G. and Degani, R., "Linguistic Approximation of Fuzzy Certainty Factors in Computerized Electrocardiograph," in *Fuzzy Computing*, M. M. Gupta and M. Yamakawa (eds.), North-Holland, pp. 243~246, 1988.
- Cohen, M. E. and Hudson, D. L., "The Use of Fuzzy Variables in Medical Decision Making," in *Fuzzy Computing*, M. M. Gupta and M. Yamakawa(eds.), North-Holland, pp. 263~272, 1988.
- Hirota, K. and Ozawa, K., "Fuzzy Flip-Flop as a Basis of Fuzzy Memory Modules," in *Fuzzy Computing*, M. M. Gupta and M. Yamakawa(eds.), North-Holland, pp. 173~183, 1988.
- Horita, K. and Ozawa, K., "Fuzzy Flip-Flop Registers," *Fuzzy Sets and Systems*, Vol. 32, pp. 139~248, 1989.
- Hirota, K. and Ozawa, K., "The Concept of Fuzzy Flip-Flop," *IEEE Trans. Sys. Man and Cybernetics*, Vol. 19, No. 5, pp. 980~997, 1989.
- Isik, C., "Inference Hardware for Fuzzy Rule-Based Systems," in *Fuzzy Computing*, M. M. Gupta and M. Yamakawa(eds.), North-Holland, pp. 185~194, 1988.
- Kaufmann, A., "Logics for Expert Systems: With Fuzzy Sets, Interval of Confidence, Probabilistic Sets and Expertons," in *Fuzzy Computing*, M. M. gupta and M. Yamakawa (eds.), North-Holland, pp. 225~242, 1988.
- Mukaidono, M., "The Presentation and Minimization of Fuzzy Switching," *Analysis of Fuzzy Information*, Vol. 1, Bezdek, J. c., pp. 213~229, 1987.
- Sultan, L., "A Formal Approach for the Organization and Implementation of Fuzzy Micro-Processor Module," in *Fuzzy Computing*, M. M. Gupta and M. Yamakawa(eds.), North-Holland, pp. 201~221, 1988.
- Symon, J. R. and Watanabe, H., "Fuzzy Logic Inference Engine Board System." *Proceedings of the International Conference on Fuzzy Logic & Neural Networks*, Iizuka, Japan, 1990.
- Togai, M. and Watanabe, H., "Expert System on a Chip: An Engine for Real-Time approximate Reasoning," *IEEE EXPERT*, Fall, 1986.
- Yamakawa, T. and Miki, I., "The Current Mode Fuzzy Logic Integrated Circuits Fabricated by the Standard CMOS Process," *IEEE Trans. on Computers*, Vol. c-36, No. 2, Feb 1986,
- Yamakawa, T., "Fuzzy Logic Circuits in Current Mode," *Analysys of Fuzzy Information*, Vol. 1, Bezdek, J. C., pp. 241~262, 1987.
- Yamakawa, T., "Intrinsic Fuzzy Electronic for Sixth Generation Computers," in *Fuzzy*

*Computing*, M. M. Gupta and M. Yamakawa (eds.), North-Holland, pp. 157~171, 1988.

15. Zbijian, L. and Hong, J., "CMOS Fuzzy Logic Circuits in Current-Mode toward Large Scale Integration," *Proc. of the International Conference on Fuzzy Logic & Neural Networks*, Iizuka, Japan, July 1990.

16. Yamakawa, T., "Stabilization of an inverted pendulum by a high-speed fuzzy logic controller hardware system," *Fuzzy Sets and Systems*, Vol. 32, No. 2, pp. 161~180, 1989.

17. 이광형, 오길록, "퍼지이론 및 응용 II 권", 홍릉과학출판사, 1991.



이 광 형

1973년 서울대학교 공과대학  
산업공학 학사  
1980년 한국 과학원 산업공학  
석사  
1982년 프랑스 INSA 진산학  
과 석사(DEA)  
1985년 프랑스 INSA 진산학

과 공학박사

1988년 프랑스 국가박사(진산학: INSA-LYON 1 디)  
1985년부터 한국과학기술대를 거쳐 한국과학기술원에서  
부교수로 근무중  
관심분야: Fuzzy systems, Expert systems, Software engi-  
neering 등



박 규 호

1973년 서울대학교 공과대학  
전자공학과 학사  
1975년 한국 과학원 전자공학  
과 석사  
1983년 프랑스 파리 VI대학  
전자공학 박사  
1983년부터 한국과학기술원 진

기 및 전자공학과에서 부교수로 근무중

관심분야: VLSI 칩 설계, 병렬처리 시스템, Vision 등