PCI 기반 병렬 퍼지추론 시스템의 설계 및 구현

Design and Implementation of PCI-based Parallel Fuzzy Inference System

이 병권, 김 종혁, 손 기성, 이 상구
Byoung Kwon Lee, Jong Hyuk Kim, Ki Sung Son, Sang Gu Lee
한남대학교 컴퓨터공학과
Dept. of Computer Engineering, Hannam University

ABSTRACT

본 논문은 대량의 퍼지 데이터를 고속으로 전송 및 추론하기 위한 PCI 기반 병렬 퍼지 시스템을 구현한다. 많은 퍼지 데이터의 고속전송을 위해 PCI 인터페이스를 사용하고, 병렬 퍼지 추론 시스템을 위한 병렬 퍼지 모듈들을 FPGA로 설계하여 PCI 타겟 코어로서 병렬로 동작하게 한다. 이러한 시스템을 VHDL을 사용하여 설계 및 구현하였다. 본 시스템은 고속의 퍼지추론을 요하는 시스템 또는 대규모의 퍼지 전문가 시스템 등에 활용될 수 있다.

Keywords: PCI bus, Parallel fuzzy inference, VHDL, FPGA

1. 서 론

퍼지논리는 0과 1의 이진 논리가 아닌, 0과 1사이의 실수 연산이 필요하기 때문에 수천개 이상의 퍼지규칙을 갖는 퍼지 추론은 고속으로 수행하기 쉽지 않다[1, 2]. 특히, 항공기나 인공위성으로부터 얻어진 화상에서 지표면의 특징을 분류하여 피복도를 작성하는 원격탐사 화상의 패턴분류 시스템과 같이 퍼지 연산이 대규모로 실행되면서 실시간성이 요구되는 시스템이나 대규모의 전문가 시스템 등에서의 처리에는 아직 해결하여야할 문제점들이 많이 남아있다[3, 4]. 또한, 지금까지 개발된 대부분의 퍼지 하드웨어들은 AND/OR(Min/Max)의 연산은 병렬로 수행하고 있지만, 퍼지규칙들 또는 전건부, 후건부에 대해서는 순차적으로 수행하고 있으므로, 이러한 부분들에 대해 병렬로 수행할 수 있는 새로운 기법에 대한 연구가 절실하게 필요하다. 따라서 퍼지규칙의 전건부 또는 후건부에 대한 변수들을 각각 병렬로 추론할 수 있는 효율적인 퍼지 병렬화 추론방법과 각각의 퍼지 소속함수에 대한 Min · Max 연산 시에 각 요소별로 병렬화 하여 고속처리를 할 필요가 있다. 최근에는 퍼지시스템을 구현할 때 FPGA를 사용하여 응용목적에 맞는 전용 퍼지 시스템으로 설계하는 경우가 많이 있다[5, 6]. 본 논문에서는 대용량 원격탐사 화상의 패턴분류 시스템에 적용할 수 있는 고속의 병렬 퍼지 시스템을 설계하고 구현한다. 대용량의 퍼지 데이터 전송을 위하여 IBM PC상의 PCI 확장 버스를 사용하고, 병렬 추론이 가능한 퍼지 추론 모듈들을 FPGA로 설계하여 PCI 타겟 코어로서 고속으로 동작할 수 있도록 한다.

2. 병렬 퍼지추론 방법론

2.1 병렬 퍼지추론 알고리즘

본 논문에서 제안하는 병렬 퍼지추론은 전건부의 퍼지변수가 동시에 여러 개의 퍼지 프로세서 모듈(FPM)에 의해 추론되고, 후건부도 모듈을 통한 병렬로 추론된다. 본 논문에서는 그림 1과 같은 알고리즘을 사용하여 퍼지추론의 병렬화를 실행한다[7, 8].

```
Let D=1
For i:=1 step 1 until r do
   begin
     d_i = max[C_ij(x)∧A_j(x)], (1 ≤ j ≤ m);
     D_j = min(D_i, d_j), (1 ≤ j ≤ m);
   end
let B(y) = 0
For h:=1 step 1 until q do
   begin
     O_kh = max[D, S^T_ik[h](y)], (1 ≤ k ≤ n);
   end
```

그림 1 병렬 퍼지 알고리즘

위와 같은 알고리즘을 사용하여 퍼지연산을 병렬로 수행하면 기존의 시리얼 방법으로 추론하던 퍼지 시스템보다 고속으로 처리할 수 있다. 기존의 방법은 간단한 퍼지 연산을 수행하면 무리가 없었지만, 대량의 퍼지데이터의 실시간 추론을 요하는 작업에 대한 퍼지 추론시 연산 횟수의 증가로 속도가 저하된다. 따라서 고속의 PCI 데이터 전송을 위한 병렬퍼지 추론 시스템을 설계하여 한 프로세서에서 수행하던 것을 여러 프로세서 모듈로 나누어 독립적으로 수행하면 가능하다. 그림 1에서 전반적인 병렬퍼지 추론 알고리즘을 보여 주고 있다.

본 논문에서는 그림 1을 기반으로 하여 다음과 같은 두 가지의 병렬처리 구조를 가지고 있다.

- 제 1병렬(Low-level)
  Min·Max 연산시 병렬(전건부 & 후건부)로 각각의 Resolution 된 값들의 연산시에 발생한다.
- 제 2병렬(Medium-level)
  퍼지 변수 사이의 병렬로 제 1병렬이 수행되는 동시에 병렬이 발생한다(전건부 & 후건부)

이와 같이, 병렬이 2곳에서 동시에 발생하므로 하나의 병렬로 처리하는 것 보다 성능은 더 우수하다.

2.2 퍼지 데이터 변환

퍼지 연산을 위한 퍼지 데이터의 변환 과정은 다음 그림 2와 같은 과정을 거쳐 메모리 및 레지스터에 저장된다. 저장은 하나의 퍼지 소속함수를 resolution 하고 이에 상응하는 퍼지값의 범위인 '0'과 '1'사이를 16등분하여, 그곳에 해당하는 높이 값을 하나에 4비트의 배열 형태의 레지스터에 저장된다. 각각의 셀들은 4비트의 퍼지값을 가지고있고 이것은 하나의 값을 형성되어, 퍼지 연산을 위한 데이터 저장을 하게된다.

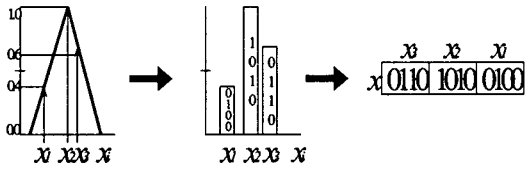

그림 2. 퍼지데이터 변환 과정

3. 병렬 퍼지 추론 시스템

3.1 데이터 입출력 인터페이스 설계

대량의 퍼지 데이터의 입·출력시 PCI 보드를 통한 보다 빠른 데이터 전송으로 퍼지 데이터를 고속 및 실시간에 처리할 수 있도록 하였다. 또한 타겟 코어 쪽에서 복잡한 퍼지연산을 병렬로 수행하여 CPU의 사용을 줄이고 보다 빠르고 효과적인 연산을 수행할 수 있다. PCI를 통하여 보내지는 데이터의 전송단위는 8bit, 16bit, 32bit 단위로 전송 가능하고 일반 병렬포트 전송 및 ISA, EISA 버스보다 빠른 132바이트(132Mbyte/sec) 데이터를 전송한다[9]. 다음의 표 1은 PCI 버스의 특성 보여주고 있다.

표 1. PCI 버스의 특성 비교

Expansion Bus	처리단위	속도
ISA	8.33Mhz, 16-bit wide	8.33Mbyte/sec
EISA	8.33 Mhz, 32-bit wide	33Mbyte/sec
PCI	33Mhz, 32-bit wide	132Mbyte/sec

본 논문에선 고속의 데이터 전송을 위한 버스로 PCI 확장 버스를 선정하고, 퍼지추론 코어와 연결하기 위해 bridge 역할로 사용되는, PCI 타겟 인터페이스 칩으로 PCI9050 칩(PLX technology)을 사용하여 고속의 데이터 인터페이스를 구현했다. 데이터 전송과정을 보면, 데이터를 센서 또는 문서로부터 받아 PCI 버스에 전달되고, 이 데이터들은 다시 PCI 인터페이스인 PCI9050 칩을 통하여 데이터를 칩의 입·출력을 메모리 맵(memory Map) I/O 방식으로 각각의 ARM(Antecedent Rule Memory), CRM(Consequent Rule Memory), FIR(Fuzzy Input Register)에 전송된다. 또한, 디코더(2×4)의 역할로 어떤 메모리와 레지스터에 퍼지 소속함수 및 퍼지 입력데이터를 저장할지를 결정하는 역할을 담당한다. 또한, SRAM을 사용하여 퍼지 소속함수를 저장하고 전건부 저장은 ARM, 후건부는 CRM에 저장한다. 마지막으로, FIR에 퍼지입력 데이터들이 저장되고 나면 start 신호에 의하여 퍼지추론 연산을 위한 모든 데이터가 준비되었음을 퍼지추론 코어에 알린다. 퍼지 코어는 4비트 단위로 퍼지 데이터를 분할하여 FPM(Fuzzy Processor Module)의 레지스터에 전달과 동시에 각각의 레지스터끼리 퍼지연산을 병렬로 연산을 수행하게된다. PCI 확장 버스와 퍼지추론 코어 메모리들과의 연결회로는 그림 3과 같다.

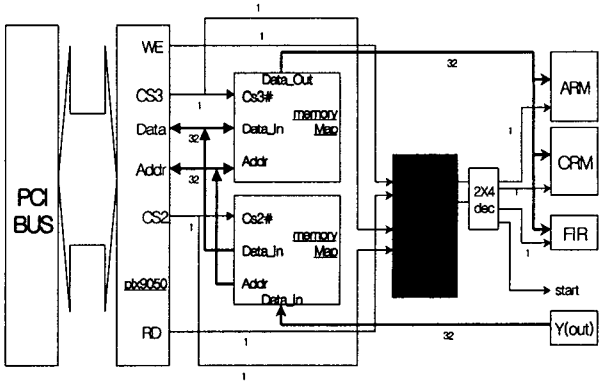

그림 3. PCI 버스와 퍼지추론 코어 메모리 연결

메모리 부분과 PCI 버스와의 연결을 위한 디코더는 다

음 표 2와 같다.

표 2. 2×4 디코더 진리표

input				count	output		state	select
we	rd	cs3	cs2		flag1	flag2		
L	H	L	H	1	0	0	write	FIR
L	H	L	H	2	0	1	write	ARM
L	H	L	H	3	1	0	write	CRM
L	H	L	H	4	1	1	start	Signal
H	L	H	L	Yn(연산결과 읽어오기)				

3.2 병렬 퍼지 추론 시스템 설계

퍼지 추론을 위한 프로세서는 전체적으로 여러 개의 모듈별로 나누어져있고, 모듈의 개수는 퍼지 입력변수와 출력변수의 최대치의 개수로 결정된다. 이 모듈들이 연결되어 전체의 병렬 퍼지 프로세서가 구성되고 각 모듈들은 전건부의 퍼지 소속함수와 입력 퍼지값이 병렬 퍼지연산을 수행해 이것의 퍼지 적합도(degree of fulfillment)를 계산 후 다시 후건부 규칙과 연산을 통하여 결과값을 만들어낸다. PFIS(Parallel Fuzzy Inference System)는 퍼지 모듈들과 Min·Max연산을 위한 FALU(Fuzzy ALU), 소속함수 저장 장치인 ARM, CRM 등으로 구성되어 있다(그림 4). 각 구성요소들은 다음과 같다.

- ▶ ARM : 전건부(Antecedent part)의 규칙 저장.
- ▶ CRM : 후건부(Consequent prat)의 규칙 저장.
- ▶ FIR : 퍼지 입력을 받아 저장하는 레지스터.
- ▶ MSR : 적합도(degree of fulfillment)저장.
- ▶ FALU: Fuzzy Min·Max수행하는 연산 장치.
- ▶ FFR : 후건부 퍼지 소속함수의 규칙과 계산하기 위한 적합도 저장 위한 레지스터.
- ▶ Yn : 최종 출력결과 값을 I/O interface를 통하여 IBM-PC 전달.

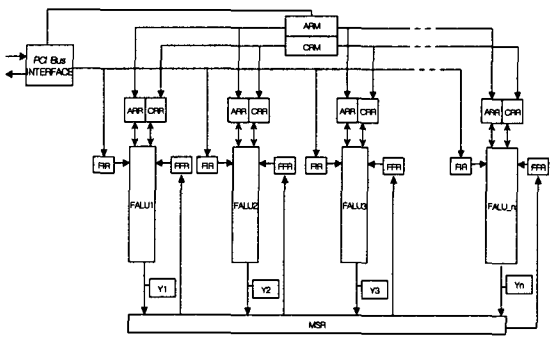

그림 4. 퍼지 추론 코어 내부연결 구조

초기화를 위하여 PCI 인터페이스로부터 각 퍼지규칙에 대하여 받아들인 퍼지 소속함수의 값을 메모리인 ARM과 CRM에 저장한다. 초기화 작업이 종료되면 센서를 통하여 들어온 퍼지 입력값(싱글톤 또는 퍼지값)과 연산하기 위하여 ARM으로부터 ARR(전건부 연산 레지스터)로 퍼지 소속함수의 값이 전송되고 후건부 연산시에도 CRR(후건부 연산을 위한 레지스터)에 전송된다. 이렇게 메모리와 레지스터로 구분한 것은 PC에서 데이터를 받아들이는 용량이 정해져 있기 때문이다. PFIS의 동작 순서는 다음과 같다.

1. I/O 인터페이스 통해 전건부와 후건부의 소속함수들의 값이 메모리에 저장된다(ARM, CRM).
2. 센서를 통하여 퍼지 입력 값이 FIR에 들어온다.
3. 각각의 FALU는 이 값들의 Min·Max 연산을 수행한다.
4. 수행된 값은 MSR(Min Store Register)에 적합도가 저장된다.
5. 적합도가 저장된 MSR값들은 후건부와 연산을 위한 FFR에 옮겨 CRM과 Min·Max 연산을 수행한다.
6. FALU가 후건부를 연산하고, 최종 출력값을 Yn에 보낸다.
7. Yn에 들어오는 값은 PCI 인터페이스를 통하여 PC에 전달되고 PC에서 비퍼지화 과정을 수행한다.

3.3 FPM(Fuzzy Processor Module)

전체 병렬 퍼지추론 코어에서 사용되는 퍼지 프로세서모듈(FPM)의 개수는 전건부 퍼지변수의 수와 후건부 퍼지변수의 최대치의 개수를 가지고 있으며, 이 퍼지모듈 하나가 전건부 또는 후건부의 퍼지변수 하나를 처리하고 또 다른 변수는 다른 퍼지 프로세서 모듈에서 동시에 병렬로 퍼지연산을 수행한다. 이 퍼지 모듈들은 전건부의 퍼지추론시 사용되고 후건부 퍼지추론 연산시에도 전건부에서 계산된 결과 값을 받아 이 모듈을 재사용 함으로써 프로세서의 효율을 최대한 높였다(그림 5).

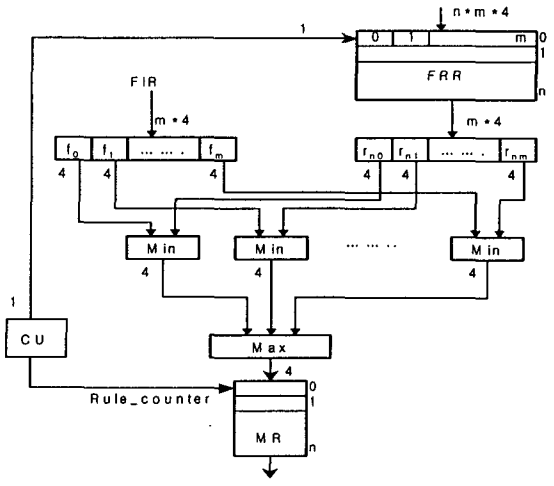

그림 5. FPM에서의 Min·Max 연산 회로

FPM은 전체 퍼지 모듈중 하나를 보여주는 것으로 FIR에서 각각이 4비트인 m개를 퍼지 입력(m×4) 값들을 받아서 퍼지소속 함수가 저장된 ARM과 Min · Max 연산을 수행한다. FRR(Fuzzy Rule Register)은 n의 규칙과 각 규칙당 m개이고 각 비트가 역시 4비트(n×m×4)로 구성되었다. 또한, CU로부터 제어 신호를 입력받아 퍼지 소속함수와 병렬로 Min 연산이 수행되고 Min 연산이 수행된 후 나온 결과의 최대값이 MR(Max Register)에 저장된다. MR도 역시 CU에 의해서 규칙별로 연산된 값들을 순서대로 Max값(n개)을 저장한다. 이 결과로 나온 값은 다른 모듈프로세서에서 계산된 Max값들과 비교하여 절단을 위한 Min연산을 수행한 후 MSR(Min Store Register) 레지스터에 적합도를 저장한다.

3.3 병렬 퍼지추론 시스템 구현

구현 환경으로는 IBM-PC (PIII-800), PLX9050 Chip, FPGA 설계도구(HBE-DTK-240), VC++6.0, Max+plus Ⅱ를 사용했다. 논문에선 IBM-PC를 이용한 FPGA 설계 툴인 Max+plus Ⅱ로 타겟 코어를 설계하고 이것을 ALTERA 칩에 다운로딩하는 방법으로 병렬처리추론 시스템 코어를 구현하여 분석했다. 실험을 위하여 VC++ 6.0을 사용해 PC와 타겟 코어 사이에 데이터 전송(8bit 및 32bit 단위의 전송)으로 전건부 퍼지 소속함수 및 후건부 퍼지 소속함수를 받고, 퍼지입력은 단일형태와 문서화 형태의 자유롭게 넣을 수 있도록 구현했다. 또한, 병렬 퍼지연산을 수행하기 위하여 입력 퍼지 데이터와 퍼지 소속함수들을 응용 프로그램인(VC++)을 이용 데이터 버스선의 용량단위(block)로 나누어 타겟 코어에 전달할 수 있고, 전송된 퍼지데이터들은 PCI의 메모리영역에 써지고 PCI 인터페이스칩인 PCI 9050칩을 통하여 퍼지 추론 코어의 전건부, 후건부 저장 메모리에 저장된다. 또한, 센서를 통하여 얻어진 입력 퍼지값 또는 단일 입력 값들이 FIR에 저장되어진다. PFIS는 전건부와 후건부의 퍼지 소속함수들을 계산하고, 그 결과 값을 다시 PCI 인터페이스를 통하여 PC에서 전달받아 비퍼지화에 사용된다. 그림 6은 이 과정을 보여 주고있다.

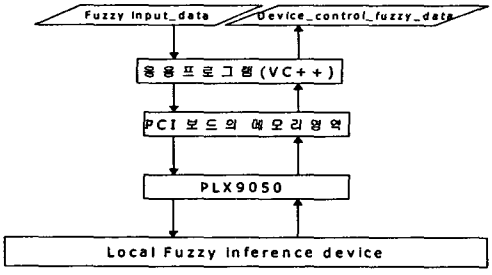

그림 6. PFIS 동작 순서도

그림 7에서는 전건부 퍼지 변수와 후건부 퍼지 소속함수 들이 메모리에 각 규칙으로 저장되고 또한, 적합도(degree of fulfillment)를 보여주는 Calc_Result 부분과 최종 계산된 결과를 표시하는 부분 Result A, B로 구성되고, 대량의 데이터는 한번에 전송이 불가능하므로 32bit 또는 8bit로 퍼지데이터를 전송할 수 있다. 또한 PCI전송 속도가 33Mhz의 속력가지고 있기 때문에 전건부 및 후건부의 퍼지규칙들을 저장시에 2가지방법으로 저장할 수 있게 설계했다. 첫 번째 방법은 임으로 퍼지데이터를 입력하는 방법이고, 두 번째 방법으로 문서단위로 퍼지 데이터를 받아들여 한꺼번에 입력하는 방법이다.

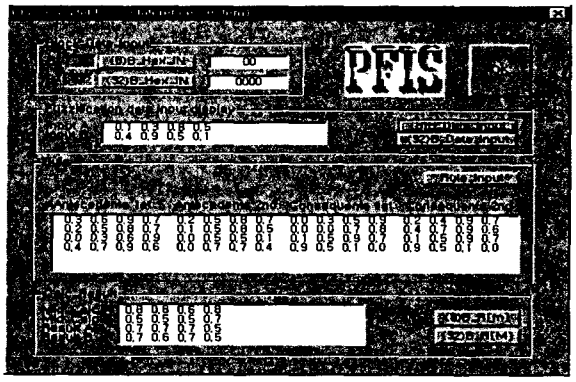

그림 7. PFIS 응용프로그램

4 실험 결과 및 분석

퍼지 프로세서 모듈(FPM)을 실험 및 분석 위한 Max+plus Ⅱ 툴의 시뮬레이션 에디터를 이용했고, Synopsys Design Analyzer이용하여 PFIS을 합성하였다. 이 시뮬레이션은 2개의 퍼지 프로세서 모듈사이에서 계산되는 시간을 측정한 결과를 보여 주고있다(그림 8).

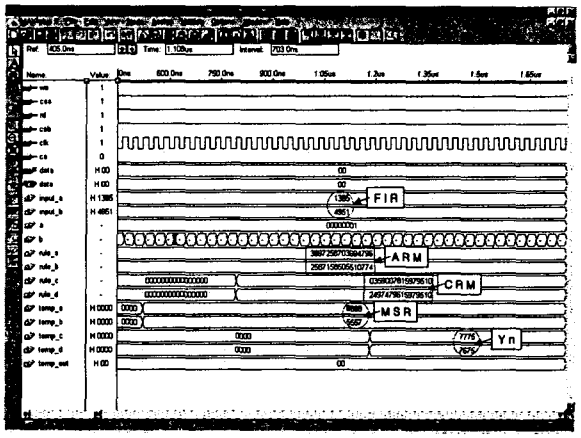

그림 8. PFIS 시뮬레이션

실험 대상 조건으로 전건부 및 후건부 퍼지규칙의 수가 4개와 입력퍼지 변수 1개가 계산되는(FPM) 시간을 측정했다. 이것은 실험으로 사용된 설계도구의 게이트의

수가 적어서 많은 규칙을 한 번에 입력이 불가능하다. 이것의 해결로 모듈단위로 코어 설계하고 각각의 모듈을 연결하면 가능하다. 이 실험에서는 작은 용량을 갖는 각각의 모듈들을 연결하여 전체적인 퍼지추론 코어를 설계했다. 또한 퍼지추론에서 시간 측정이 문제이므로 퍼지값을 변환된 데이터와 무작위로 뽑은 데이터를 넣어 분석했다. 그림 8에서 모듈사이가 같은 시간에 MSR의 값이 나타나는 것을 볼 수 있다. 그림 9과 그림 10은 Synopsys Design Analyzer에서 PFIS 와 하나의 퍼지모듈인 FPM을 합성한 것을 보여 주고 있다.

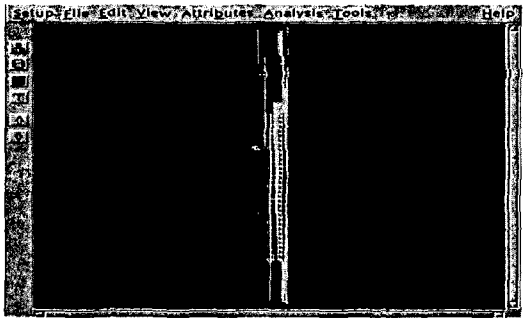

그림 9. PFIS의 Synopsys 합성

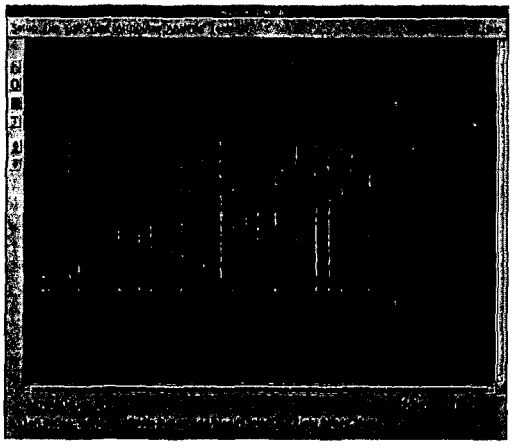

그림 10. FPM Synopsys 합성

퍼지 소속함수의 요소 수에 따른 본 논문에서 제안한 방식과 일반적인 순차적 퍼지추론 시스템의 성능평가를 보면 표 4와 같다.

표 3. PFIS & SFIS의 성능평가(단위: ns)

System / Resolution N	DT	MT	PFIS(L)	SFIS(L)	PFIS(M)	SFIS(M)
4	20	90	170	440	250	1840
8	20	90	250	880	410	7200
16	20	90	410	1760	730	28480
32	20	90	730	3520	1370	113280
64	20	90	1370	7040	2650	451840
128	20	90	2650	14080	5210	903680
256	20	90	5210	28160	10330	7214080
512	20	90	10330	56320	20570	28846080

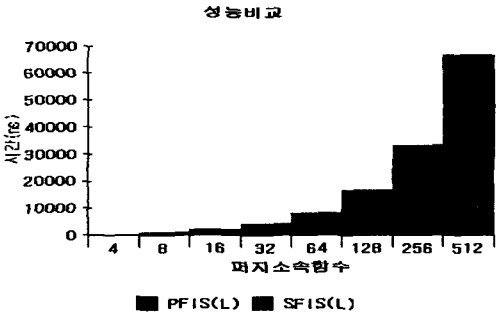

그림 11. PFIS & SFIS

그림 11은 본 논문에서 제안한 PFIS와 일반적인 SFIS의 수행시간의 성능 비교를 나타낸 것으로, 처음에는 두 방법이 수행시간에 별 차이를 보이지 않지만, 퍼지 소속함수의 수가 증가함에 따라 현저한 차이를 보이고 있다. 즉, PFIS에서 퍼지 소속함수의 수에 따라 수행시간이 조금씩 증가하는 것은 퍼지 소속함수의 전송시간에 따른 데이터의 전송지연이 발생하기 때문이다.

5. 결론 및 향후 연구

본 논문에서는 대량의 퍼지 데이터를 위한 PCI 기반 병렬 퍼지추론 시스템을 설계하고 구현하였다. 많은 퍼지 데이터 고속 전송을 위해 IBM-PC의 PCI 확장 버스를 사용하였으며, 병렬 퍼지추론 모듈들은 FPGA로 설계하여 PCI 타겟 코어로서 동작할 수 있도록 하였다. 이러한 FPGA로 설계된 병렬 퍼지추론 모듈의 특징은 (1) 하나의 전건부 또는 후건부의 변수를 처리할 때 각각의 요소들의 병렬처리(각각의 Min · Max 연산시), (2) 전건부 또는 후건부 변수들(FPM)병렬로 처리한다.

본 시스템은 기존의 소프트웨어적인 접근 방법, 전용 퍼지 하드웨어 시스템, 또는 단순한 레벨의 병렬구조보다 높은 성능 특성을 갖는다.

이러한 PCI 기반 병렬 퍼지추론 시스템은 고속 처리 및 실시간을 요구하는 위성탐사에 활용할 수 있다. 향후 연구 과제로 본 연구에서 사용된 SRAM 대신에 한번에 128 비트의 데이터를 처리가 가능한 고속처리 메모리인 RDRAM 사용으로 퍼지데이터의 저장으로 고속 입출력이 가능한 병렬퍼지 추론 시스템 개발에 대한 연구가 필요하고, 본 PFIS 시스템을 활용한 영상처리 패턴 인식할 수 있는 실제 응용프로그램 개발로 실용성 평가작업이 수행되어야 할 것이다.

감사의 말

본 연구는 한국과학재단 목적기초연구
(2000-1-30300-007-2) 지원으로 수행되었음.

참고문헌

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

[2] A. Costa et al., "Hardware Solutions for Fuzzy Control," *Proc. of the IEEE*, Vol. 83, No. 3, pp. 422-434, Mar. 1995.

[3] M. J. Patyra et al., "Hardware implementations of digital Fuzzy logic controllers," *Information Sciences*, Vol. 113, pp. 19-54, 1999.

[4] G. Ascia et al., "VLSI Hardware Architecture for Complex Fuzzy Systems," *IEEE Tr. on Fuzzy Systems*, Vol. 7, No. 5, pp. 553-570, Oct. 1999.

[5] D. Hung, "Dedicated Digital Fuzzy Hardware," *IEEE Micro*, Vol. 8, pp. 31-39, Aug. 1995.

[6] J. J. Blake et al., "The implementation of fuzzy system, neural networks and fuzzy neural network using FPGAs," *Information Sciences*, Vol. 112, pp. 151-168, 1998.

[7] 이 상구, "퍼지 정보처리를 위한 효율적인 병렬 퍼지 아키텍처의 설계", 한국정보과학회 논문지(C), Vol. 4, No. 4, pp. 567-574, 1998. 8

[8] Sang Gu Lee and K. Akizuki, "Design of Effective Parallel Fuzzy Architecture for Fuzzy Information Processing," *Trans. of IEEJ*, Vol. 118, No. 7/8, pp. 1190-1195, Aug. 1998.

[9] PCI 9050-1 Data Book, PLX technology, 1998.