TSK퍼지 시스템의 ASIC 설계

ASIC design of TSK-Fuzzy system

김태성, 강근택, 이원창
부경대학교 전자공학과
Taesung Kim, Guentaek Kang, Wonchang Lee
Pukyong National Univ., Pusan, Korea

Abstract - 퍼지 시스템은 비선형 시스템을 해석하고 제어기 설계 등에 많이 이용되고 있으나 대부분의 그 구현은 PC나 웍스테이션의 프로그램에 의존하고 있다. 고속의 동작을 요구하는 시스템이나 소형 시스템에는 전용 프로세서의 사용이 필요하다. 본 논문에서는 여러 퍼지 시스템 중에서 적은 규칙수로도 효과적인 성능을 나타내고 결론부가 선형식으로 표현되어 ASIC을 이용한 하드웨어화가 용이한 형태를 가진 TSK퍼지 추론 프로세서를 FPGA로 구현한다. ASIC의 설계는 Top-down 방식을 이용하여 전체구성은 Schematic을 이용하고 기능블록은 VHDL로 기술한다. TSK퍼지 추론의 연산은 전제부와 결론부를 병렬연산함으로써 고속처리를 구현하고 이에 필요한 제어부를 설계하였다. 또한 하드웨어 구현을 위해 실수연산을 이산화된 연산으로 바꾸고 이에 따른 나누기 연산자를 구현하였다.

I. 서론

퍼지 시스템은 복잡한 비선형 시스템의 해석 및 제어기 설계 등에 많이 이용되고 있고 다양한 퍼지 추론의 형태가 존재한다. 그러나 퍼지 모델의 추론을 구현하는 방법으로 현재까지는 대부분 소프트웨어적인 방법으로 PC나 웍스테이션의 범용 프로세서를 이용에 의존하고 있다. 범용 프로세서를 이용할 경우 계산 시간이 매우 많아져 빠른 속도를 요구하는 시스템에서는 적용하기 어려우며 소형의 시스템에서는 내장할 수가 없다. 따라서 퍼지 모델의 추론을 하드웨어적인 방법으로 처리하여 많은 입출력 변수를 가지고 빠른 처리 시간과 다입력에 대한 병렬 처리를 할 수 있는 퍼지추론 프로세서의 개발이 요구된다.

본 논문에서는 여러 형태의 퍼지 모델의 추론법 중에서 결론부가 선형식으로 표현되고 전제부의 퍼지 집합이 비교적 적은 수의 간단한 선형식으로 표현이 가능한 TSK퍼지 추론을 위한 프로세서를 FPGA를 이용하여 설계하였다. TSK퍼지는 결론부가 선형식으로 구성되어 하드웨어로의 구현이 용이한 형태를 가지고 있으며 규칙의 수가 적어 연산에 필요한 계수의 저장 공간이 줄어드는 이점이 있다.

본 논문은 본론의 1장에서 TSK퍼지 모델의 추론을 설명하고, 2장에서 하드웨어화를 위한 변형된 추론법과 실제구현에 대해 설명하고 결론을 맺는다.

II. 본론

1. TSK퍼지 모델

TSK퍼지 추론법[3]은 복잡한 시스템에 대해서도 해석적인 지식이 필요 없이 시스템의 입출력 데이터만으로 퍼지모델의 작성이 가능하고 TSK퍼지 모델로부터 TSK퍼지 제어기를 직접 구할 수 있는 특징이 있다. TSK퍼지 모델

은 다음과 같이 결론부가 선형식인 퍼지 규칙 들로 구성되어 선형 제어 이론의 적용이 가능하다. 퍼지규칙은 식(1)과 같다.

$$M^i: \text{ if } z_1 isF_1^i,\ z_2 isF_2^i \cdots z_m isF_m^i \quad then\ y^i = a_0^i + a_1^i x_1 + \cdots + a_n^i x_n \tag{1}$$

출력 y와 i번째 규칙의 적합도 $w^i(z)$는 다음의 식(2)와 (3)으로 구한다.

$$y = \sum_{i=1}^{m} w^i(z) y^i / \sum_{i=1}^{r} w^i(z) \tag{2}$$

$$w^i(z) = \prod_{j=1}^{m} F_j^i(z_j) \tag{3}$$

TSK 퍼지 모델에서 전제부 퍼지집합의 멤버쉽함수는 <그림1>과 같은 세 가지 형태의 구분 선형함수로 표현할 수 있다[4].

<그림1> 퍼지집합의 형태

입력 x에 대한 <그림1>의 퍼지집합의 멤버쉽값을 $A_1(x)$, $A_2(x)$, $A_3(x)$라고 하면 식(4)와 같이 나타내어진다[4].

$$A_1(x) = 0.5 - (|x-p_1| - |x-p_2|)/(p_2-p_1) \tag{4a}$$

$$A_2(x) = (|x-p_1| - |x-p_2|)/(p_2-p_1) - (|x-p_3| - |x-p_4|)/(p_4-p_3) \tag{4b}$$

$$A_3(x) = 0.5 + (|x-p_3| - |x-p_4|)/(p_4-p_3) \tag{4c}$$

2. 프로세서 구현의 고려 사항

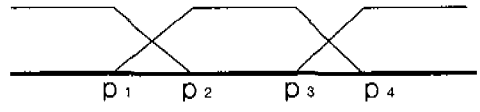
TSK 퍼지 시스템에서 멤버쉽값의 범위는 0에서 1 사이의 실수를 사용한다. 실수연산은 개별 연산자의 구현이 복잡하고 대상 프로세서에서 많은 게이트를 차지하므로 ASIC 구현이 어려울 뿐만 아니라 계산시간이 길어지게 된다. 실수 연산의 복잡함과 회로의 단순화를 위하여 본 논문에서는 정수형 연산만을 사용한다. 정수형 연산을 사용하기 위하여 실수를 해당 구간 내에서 8비트로 이산화한다. 즉, 전제부 퍼지집합의 영역이 256등분되고 멤버쉽값도 256등분되어 계산이 되며 결론부도 같은 식으로 변환되어 계산된다. 그리고 최종의 결과값도 8비트의 값이 출력되고 구성한 시스템에서 적절한 D/A변환과 증폭을 해서 전체 시스템을 구성하게 된다. 식(4)에서의 멤버쉽값 계산을 위해서는 나누기 연산자가 필요하게 되므로 이산화된 나누기의 몫을 계산하여 퍼지값을 계산할 수 있는 새로운 정수형 나누기 연산자[5]를 구현하였다. 프로세서 전체적으로는 8비트의 데이터 버스를 사용하지만 연산의 정밀도를 유지하기 위해 각 연산 블록에서는 필요한 정도의 비트 확장을 하여 계산하며 부호없는 정수형 연산을 사용하고 있다. 앞에서 제시한 멤버쉽값 계산 식(4a), (4b), (4c)는 이산화된 나누기 연산자 DIV를 이용하여 식(5)와 같이 변경하여 이산화된 멤버쉽값 $\overline{A_1}(x)$, $\overline{A_2}(x)$, $\overline{A_3}(x)$를 구한다.

$$\overline{A_1}(x) = 128 - DIV((|x-p_1| - |x-p_2|), (p_2-p_1)) \tag{5a}$$

$$\overline{A_2}(x) = DIV((|x-p_1| - |x-p_2|), (p_2-p_1)) - DIV((|x-p_3| - |x-p_4|), (p_4-p_3)) \tag{5b}$$

$$\overline{A_3}(x) = 128 + DIV((|x-p_3| - |x-p_4|), (p_4-p_3)) \tag{5c}$$

실수연산을 배제함으로써 사용해야 하는 나누기 연산자는 전체 ASIC설계에서 대상 칩내에 많은 용적을 차지한다. TSK퍼지 모델의 출력을 계산하는 식(2)에서 나누기 연산을 없앰으로서 칩을 소형화시키고 계산 속도를 향상시키기 위해서 앞에서 서술한 퍼지집합의 형태를 전제부 변수의 퍼지집합이 <그림2>와 같이 퍼지집합의 끝점이 서로 일치하도록 그 형태를 제한하는 전제부 퍼지집합의 분할을 제안한다.



<그림2> 제안된 퍼지집합

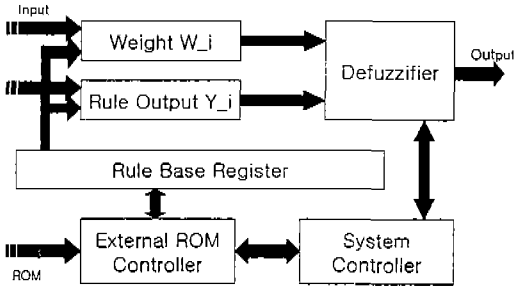
<그림2>와 같이 전제부 변수 형태를 제한하면 식(2)의 분모 부분이 항상 1이 되므로 출력 y는 다음과 같이 간략한 형태로 표현이 된다.

$$y = \sum_{i=1}^{m} w^i(z)\, y^i \qquad (6)$$

3. TSK퍼지연산 프로세서의 설계

본 논문에서 구현된 TSK 퍼지 프로세서는 최대 10개의 퍼지 규칙을 설정할 수 있고, 각 규칙에서 전제부 퍼지집합은 3개를 가질 수 있다. 5개의 입력과 1개의 출력을 가지고, 퍼지규칙을 표현하기 위해 사용되는 전제부의 네 점 P_1, P_2, P_3, P_4와 결론부 계수 a_0, a_1, a_2, a_3, a_4, a_5는 외부 ROM에 저장되며 모두 8비트로 이산화되어 있다.

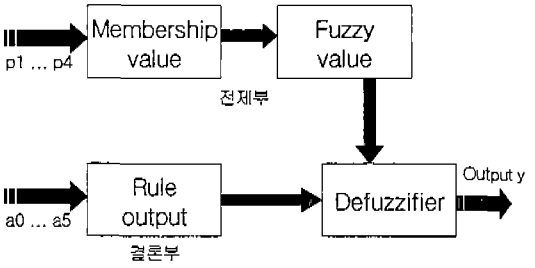
구현을 위하여 사용된 Xilinx사의 FPGA인 XCV400HQ240[1]를 사용하며 설계를 위한 툴은 Xilinx Foundation Series 2.1i[2]를 사용하였다. 설계 방식은 Top-down 방식[6]을 취하여 Schematic과 VHDL을 혼용하여 사용하였다. 먼저 Schematic 상에서 개념적인 기능블록을 선언하여 전체 시스템의 구성을 표현하고, 각 기능블록의 동작에 대한 설계는 VHDL을 사용하여 기술하였다. 구현된 전체 블록 다이어그램은 아래의 <그림3>과 같다.



<그림3> 전체 블록 다이어그램

전체 구성은 크게 둘로 나누어 TSK퍼지 연산부와 프로세서 제어부로 나눌 수 있고, 퍼지 연산부는 전제부와 결론부가 병렬 연산으로 처리되어 비퍼지화부에서 최종 출력 y를 출력한다. 프로세서 제어부는 Rule-base 레지스터와 외부 ROM 제어부, 시스템 제어부로 구성되어 프로세서가 원활한 동작을 하도록 제어한다.

TSK 퍼지 연산부는 아래의 <그림6>과 같이 연산의 특징과 순서에 따라 4개의 블록으로 나누어 설계하였다.



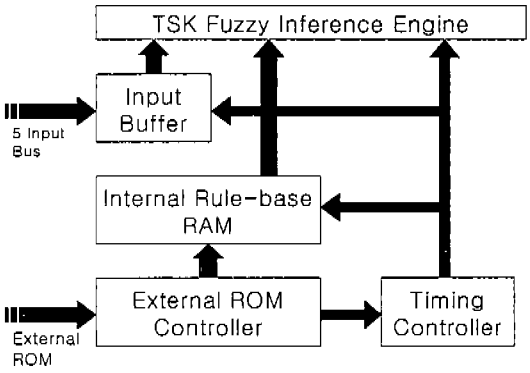
<그림4> 퍼지 연산부 블록 다이어그램

전제부의 멤버쉽값 블록은 <그림1>에서 보여지는 전제부 퍼지집합의 세 가지 형태중 하나를 선택하고 p1, p2, p3, p4의 값과 프로세서로 입력되는 x_i값을 입력받아 식(5)를 계산하고 퍼지값 계산 블록으로 값을 넘겨준다. 퍼지값 계산 블록은 멤버쉽값을 받아서 식(3)에 의해 각 규칙의 가중치 w^i를 계산한다.

결론부는 프로세서 입력 x_i와 결론부 계수를 입력받아 각 규칙의 출력 y^i를 계산하고, 마지막으로 비퍼지화 블록에서 전제부의 w^i와 결론부의 y^i로 식(6)을 이용하여 TSK 퍼지 모델의 출력 y를 계산한다.

프로세서 제어부는 시스템 전체의 초기화와 초기화시에 외부 ROM에 시스템의 특성에 따라 미리 계산되어 저장된 Rule 계수를 ASIC 내부 RAM[8]으로 읽어 오는 작업을 수행한다. 프로세서 제어부는 타이밍 제어부[7]와 Rule-base RAM과 Rule-base를 저장하고 있는 외부 ROM의 제어부로 구성되어 있다. 타이밍 제어부는 TSK퍼지 추론부와 입력값, 내부 RAM의 읽기와 연산 타이밍을 조정해 주는 역할을 한다. 각 블록은 연산종료 신호를 발생하고 정확한 연산 및 읽기 타이밍을 맞추도록 하는 동기신호를 발생한다. 외부 ROM 제어부는 연산에 필요한 전제부의 네 점과 전제부 퍼지집합의 형태를 나타내는 인덱스와 결론부의 여섯 개의 계수를 프로세서 외부의 ROM에서 읽어와서 내

부 RAM에 저장하는 역할을 하며, 초기화시에 읽어오는 동작을 한번 수행한다.



<그림5> 제어부 블록 다이어그램

Rule-base는 외부 ROM에 사용자가 적용하는 시스템에서 설계된 모델에 따라 전제부 퍼지집합과 결론부 계수가 저장되고 프로세서 동작 중에 내부 RAM으로 읽혀진다.

Rule-base 전체의 구조는 <그림6>과 같은 형태가 된다. 각 규칙의 시작은 00으로 나타내고 각 규칙의 마지막은 FF로 나타낸다. 사용하지 않는 부분은 모두 00으로 채워진다. 각 규칙은 24바이트로 구성되고, 10개의 규칙을 사용함으로 총 240바이트를 사용하게 된다.

규칙 1 (24 byte)		규칙 10 (24 byte)
00(시작)		00(시작)
결론부 (7 byte)		결론부 (7 byte)
전제부1 (5 byte)		전제부1 (5 byte)
전제부2 (5 byte)		전제부2 (5 byte)
전제부3 (5 byte)	. . .	전제부3 (5 byte)
FF(끝)		FF(끝)

<그림6> Rule-base 전체 구조

Ⅲ. 결론

본 논문에서는 FPGA를 이용하여 TSK퍼지 추론을 위한 프로세서를 설계하였다. 하드웨어화의 용이를 위하여 실수값으로 표현되던 값들을 8비트로 이산화시키고 이에 필요한 식을 제안하였고 새로운 나누기 연산자를 설계하여 적용하였다. 또한, 전제부 퍼지집합의 형태를 제한하여 비퍼지화 연산을 간단히 하여 ASIC 설계의 하드웨어화를 위한 FPGA 구현에서 나타나는 과다한 용적 문제를 고려하였으며 고속 연산 처리를 위해 TSK퍼지 추론을 위한 전제부와 결론부의 병렬연산이 되도록 하였다.

범용 마이크로 프로세서의 특징인 외부 ROM을 사용하여 계수를 저장함으로써 구현된 TSK 퍼지 추론 프로세서가 특정한 영역에서만 사용되는 것을 방지하고 다양한 영역에서 범용성을 유지하며 특정 시스템에 구애받지 않고 간단히 새로운 시스템에 적용하도록 하였다.

Ⅳ. 참고문헌

[1]Xilinx, "The Programmable Logic Data Book", version 1.03, PP.4,1-4,49. 1996

[2]Xilinx, "XACTstep Foundation", version 1.03, PP.1-128, 1996

[3]Li-Xin Wang, "A Course in Fuzzy Systems and Control", Prentice-Hall International, Inc., PP.265-276

[4]고경천, "퍼지이론을 이용한 Robot Manipulator의 역기구학 제어에 관한 연구", 부산수산대학교, PP.9-10, 1992. 11

[5]K. C Chang, "Digital Systems Design with VHDL and Synthesis", Computer Society, PP.445-465

[6]박세현, "VHDL에 의한 디지털 컴퓨터 설계와 구현", 도서출판 그린, PP.385-404, 1999. 11

[7]김태성, 이진희, 강근택, 이원창, "회전체 신호처리 시스템의 ASIC 설계", 대한전자공학회 부산·경남지부 춘계 학술대회 논문집, 1999, 6.

[8]David Van Den Bout, "FPGA DESIGN 이론 및 실습" 홍릉과학출판사, PP.329-397, 2000. 9