2006년도 대한전자공학회 하계종합학술대회 제29권 제1호

연결 단어 음성인식을 위한 하드웨어 아키텍쳐 및 FPGA 구현

김용*, 정홍**
포항공과대학교 전자전기공학과

A hardware architecture of connected speech recognition and FPGA implementation

Yong Kim*, Hong Jeong**
Dept. EEE, POSTECH
E-mail : *ddda@postech.ac.kr, **hjeong@postech.ac.kr

Abstract

In this paper, we present an efficient architecture for connected speech recognition that can be efficiently implemented with FPGA. The architecture consists of newly derived two-level dynamic programming (TLDP) that use only bit addition and shift operations. The advantages of this architecture are the spatial efficiency to accommodate more words with limited space and the computational speed from avoiding propagation delays in multiplications. The architecture is highly regular, consisting of identical and simple processing elements with only nearest-neighbor communication, and external communication occurs with the end processing elements.

I. 서론

현재 사용되고 있는 음성 인식 시스템은 소프트웨어 중심이 대부분이다. 그러나 실제 음성 인식 시스템이 실생활에 응용되기 위해서는 칩으로 구현이 되는 하드웨어의 개발이 필요하다. 음성인식 시스템이 하드웨어로 구현된 연구가 기존에 개발되었다. [1-3]의 논문들은 지금까지 소개된 히든 마코프 모델(Hidden Markov Model: HMM) 기반은 하드웨어 음성인식 시스템이다. 그러나 현재까지 하드웨어로 구현된 경우는 모두 고립 단어를 인식하는 음성인식기이다. 즉 인식하고자 하는 단위가 단어로 한정되어 인식하고 하는 단어를 중심으로 학습을 하고 이를 인식하는 것이다.

본 논문은 고립 단어가 아닌 연결 단어(connected word)를 인식하는 음성인식 시스템을 제시한다. 이때 기존에 있는 음성 인식 알고리즘 중 가장 많이 사용되고 있는 HMM 와 두 개의 레벨을 가지는 동적 프로그램밍(Two-Level Dynamic Programming: TLDP)을 바탕으로 하드웨어로 구현하기 위한 병렬 배열 구조를 가는 하드웨어 아키텍쳐를 설계하였다.

II. 이론적 배경

연결단어(connected word)를 인식하는 음성인식 시스템을 구현하기 위해서는 단어의 경계점을 찾아야 한다. 즉 단어의 경계점만 찾게 되면 기존의 고립단어 인식과 같이 HMM 알고리즘을 이용하여 인식을 할 수 있다. 그러나 단어의 경계점을 찾은 일은 쉽지 않다. 기존에 연결단어를 인식하기 위해서 나온 알고리즘으로 TLDP algorithm, level building algorithm, one-pass algorithm 등이 있지만 본 논문에서는 하드웨어로 구현이 적합한 TDLP 을 이용하였다.

.

TLDP 는 이름에서도 알 수 있듯이 두 개의 레벨의 동적 프로그래밍으로 이루어져 있다. 우선 첫 번째 레벨에서는 임의의 시작점과 끝점에서 레퍼러스 패턴과 테스트 패턴의 매칭 코스트를 다음과 같이 계산한다.

$$\hat{D}(v,s,e) = \min_{w(m)} \sum_{m=s}^{e} d(\vec{t}(m), \vec{r}_v(w(m)) \quad (1)$$

여기서 s(1≤s≤M)는 시작점을, e(1≤e≤M, e>s)는 끝점을 R_v(1≤v≤V)는 V 개의 인식하고 하는 레퍼런스 단어 중 v 번째 단어의 패턴을 나타낸다. M 은 전체 프레임 크기를 나타낸다. 위 식은 (s,e) 구간에서 테스트 패턴과 특정 레퍼런스 패턴의 매칭 값을 나타낸다. 값에서 모든 레퍼런스 중 가장 매칭 코스트가 작은 값을 찾으면 다음과 같이 쓸 수 있다.

$$\tilde{D}(s,e) = \min_{1\le v\le V}[\hat{D}(v,s,e)] \quad (2)$$

$$\tilde{N}(s,e) = \arg\min_{1\le v\le V}[\hat{D}(v,s,e)] \quad (3)$$

두 번째 레벨에서는 첫 번째 레벨에서 구한 $\tilde{D}(s,e)$을 이용하여 전체 프레임과 가장 매칭이 잘 되는 단어의 배열을 구하는 것이다. 여기서 전체 프레임이 몇 개의 단어로 이루어져 있는지도 매칭 과정을 통해 결정하게 된다. 첫 번째 패턴부터 e 번째 패턴까지의 l 개의 레퍼런스와 최적의 매칭 코스트를 $\bar{D}_l(e)$ 라고 정의하면 다음과 같이 쓸 수 있다.

$$\bar{D}_l(e) = \min_{1\le s<e}[\tilde{D}(s,e) + \bar{D}_{l-1}(s-1)] \quad (4)$$

Ⅲ. TLDP 의 하드웨어 구조

그림 1 은 전체 시스템의 블록 다이어 그램이다. 본 하드웨어 구조는 크게 첫 번째 동적 프로그램밍 (first level dynamic programming)부분과 두 번째 동적 프로그램밍 (second level dynamic programming)부분으로 나누어져 있다. 첫 번째 동적 프로그램밍 부분에서는 입력으로 HMM 의 parameter 인 A, B 와 단어의 인덱스가 들어오면 첫 번째 레벨에서 $\hat{D}(v,s,e)$ 을 계산하여 메모리로 내보낸다. 이와 같은 과정은 모든 레퍼런스 단어를 가지고 반복적으로 수행한다. 메모리에서는 첫 번째 레벨에서 들어오는 값 중 가장 작은 값을 저장한다. 즉 메모리에는, $\tilde{D}(s,e)$가 저장되어 있고 이는 두 번째 동적 프로그램밍 부분으로 보낸다.

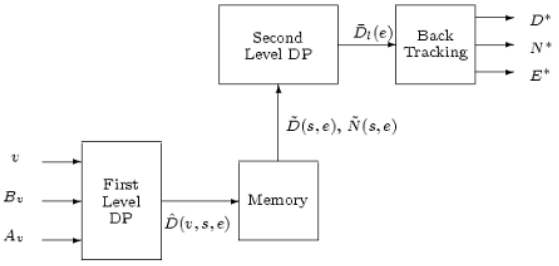

그림 1 테스트 플레이어 동작 화면 예

두 번째 동적 프로그램밍 부분에서는 $\bar{D}_l(e)$ 를 계산하고 이를 바탕으로 back tracking 과정을 통해 음성인식의 결과인 단어의 배열을 구하게 된다. 각 레벨의 동적 프로그램밍 구조는 병렬 배열 구조를 가지고 있어 칩의 용량에 따라 확장이 용이하고 칩을 여러 개 사용하여 성능을 향상 시킬 수 있는 장점을 가지고 있다.

Ⅳ. 구현

본 논문의 시스템은 33MHz의 클락 스피드로 동작하는 FPGA(Xilinx Vertex-Ⅱ XC2V8000)으로 구현하였다. 전체 칩은 VHDL으로 코딩을 하였으며, ModelSim의 프로그램을 통해 시뮬레이션을 수행하였다. FPGA는 Pentium 4, 3.06GHz processor를 가지는 PC와 PLX9656 PCI 칩으로 최대 클락 주파수 66MHz로 통신하고 있다.

참고문헌

[1] F.L. Vargas, R.D.R. Fagundes, and D.B. Junior, "A FPGA-based Viterbi algorithm implementation for speech recognition systems." ICASSP '01, vol.2, pp.1217 - 1220, 2001.

[2] Jer Min Jou, Yeu-Horng Shiau, and Chen-Jen Huang, "An efficient VLSI Architecture for HMM-Based Speech Recognition." ICECS 2001. vol.1, pp.469-472, 2001.

[3] S. Yoshizawa, Y. Miyanaga, and N. Wada, "A Low-power VLSI Design of an HMM Based Speech Recognition System." MWSCAS-2002. vol.2 pp.II-489 - II-492, 2002.