TMS320C2000 계열 DSP를 이용한 단일칩 음성인식기 구현

Implementation of a Single-chip Speech Recognizer Using the TMS320C2000 DSPs

정 익 주*
Ik-Joo Chung

ABSTRACT

In this paper, we implemented a single-chip speech recognizer using the TMS320C2000 DSPs. For this implementation, we had developed very small-sized speaker-dependent recognition engine based on dynamic time warping, which is especially suited for embedded systems where the system resources are severely limited. We carried out some optimizations including speed optimization by programming time-critical functions in assembly language, and code size optimization and effective memory allocation. For the TMS320F2801 DSP which has 12Kbyte SRAM and 32Kbyte flash ROM, the recognizer developed can recognize 10 commands. For the TMS320F2808 DSP which has 36Kbyte SRAM and 128Kbyte flash ROM, it has additional capability of outputting the speech sound corresponding to the recognition result. The speech sounds for response, which are captured when the user trains commands, are encoded using ADPCM and saved on flash ROM.

The single-chip recognizer needs few parts except for a DSP itself and an OP amp for amplifying microphone output and anti-aliasing. Therefore, this recognizer may play a similar role to dedicated speech recognition chips.

Keywords: Single-Chip speech recognizer, TMS320C2000 DSPs

1. 서 론

최근 들어 음성인식 기술의 실질적인 활용 및 상용화가 어느 정도 가시화되면서 지난 수십 년간 지속되어온 음성인식 연구는 어느 정도 그 성과가 나타나고 있다. 특히, 반도체 기술의 획기적인 발전으로 고성능의 마이크로프로세서와 고용량의 메모리를 저렴한 가격에 사용할 수 있게 되면서 연구실 및 기업에서 얻어진 성과물들이 속속 상용화되고 있다.

음성인식 시스템은 그 규모에 따라 크게 두 가지 영역으로 나누어진다. 첫 번째는 연산 성능이나 메모리와 같은 리소스의 제한이 거의 없는 호스트 서버 기반의 인식기를 이용하는 응용 분야와 두 번째로는 리소스에 제한을 받는 임베디드 시스템 기반의 인식기를 이용하는 응용 분야이다. 전자의 경우는 대용량 인식을 위주로 하는 전화망 또는 VoIP 기반의 음성인식 응용이 대표적인 예이며, 후

* 강원대학교 전기전자공학부

자의 응용 분야로는 단말기에 내장된 전용 하드웨어나 마이크로프로세서를 기반으로 한, 휴대폰이나 네비게이션과 같은 이동형 단말기에서의 중소용량 음성인식 응용을 들 수 있다. 그러나 최근 임베디드 시스템에 사용되는 CPU의 성능이 향상되고, 메모리 용량도 대폭적으로 늘어나면서 리소스의 제한이 상당히 완화되었으며, 또한 서버 급에서 사용되던 운영체제와 유사한 운영체제가 적용되면서 임베디드 시스템에서도 과거 서버 급에서 수행될 수 있었던 대용량 인식기술이 적용되고 있다. 예를 들어 최근 내장된 CPU를 이용하여 수십만 단어 인식이 가능한 네비게이션 시스템이 개발되고 있는 것이 그 한 예이다. 따라서 사용 가능한 리소스 규모에 따른 인식기의 분류는 다소 애매해졌다 [1][2].

한편, 임베디드 음성인식기는 앞에서도 언급한 바와 같이 전용 하드웨어와 내장된 CPU를 이용하는 두 가지 방식이 있을 수 있는데, 현재는 대부분 내장된 CPU를 이용하여 소프트웨어적으로 구현하는 방식을 채택하고 있다. 그 이유로는 전용 하드웨어를 이용할 경우에는 추가의 제조비용이 발생할 뿐만 아니라, 전용 하드웨어를 이용하여 대용량의 복잡한 음성인식 기술을 구현하는 데는 한계가 있기 때문이다. 그럼에도 불구하고 전용 하드웨어 방식의 경우는 기존의 시스템에 영향을 주지 않으면서 음성인식 기능을 추가할 수 있는 장점이 있기 때문에 향후 가격이 저렴해지고, 좀더 발전된 음성인식 기술이 적용되면 널리 사용될 것으로 예상된다.

전용 하드웨어 방식의 음성인식기는 아직까지는 널리 활용되고 있지 않으며, 따라서 이를 제조 판매하는 회사도 매우 적다. 현재 시장에 출시되어 있는 전용 하드웨어는 음성인식 칩과 DSP를 기반으로 한 하드웨어 모듈의 두 가지 형태가 있는데, 음성인식 칩의 경우는 가격이 저렴하다는 장점이 있는 반면 기능이 매우 제한적이고, 하드웨어 모듈의 경우는 근본적으로는 소프트웨어적으로 구현되므로 사용되는 DSP에 따라서는 고성능의 인식기 구현이 가능하나 가격이 비싸기 때문에 매우 제한적으로 사용되고 있다. 한편, 음성인식 칩의 경우는 아직 큰 시장을 형성하고 있지는 않지만, 저렴한 가격을 앞세워 장난감, 가전제품 등에 적용되기 시작하면서 나름대로 점차 활용 영역을 넓혀가고 있다.

음성인식 칩은 화자종속 기술을 적용하고 있는 것이 일반적이다. 물론 화자 독립 기술을 적용하는 것이 가능하기는 하지만, 이럴 경우 명령어를 위한 별도의 음성 데이터 처리 과정을 필요로 하거나, 가변어휘 기술을 적용한다고 하여도 음성인식 칩을 사용하기 위해서는 칩 제조사에 음성인식 칩에 해당 명령어가 포함되도록 의뢰하여야 한다. 이는 음성인식 칩을 적용하기 위해서는 여타의 부품처럼 원하는 수량을 일반적으로 구매하여 사용하기 어렵고, 칩 제조사에 의뢰하여 원하는 명령어가 포함된 음성인식 칩을 일정 수량(최소 물량) 구매해야하기 때문에 소량 생산하는 제품에 음성인식 기능을 추가하는데 어려움이 있다. 요약하면, 음성인식 칩의 경우는 저렴하다는 가격적인 장점이 있는 반면, 소량 생산하는 제품에 적용하기에는 적합하지 않다. 따라서 현재는 음성인식 칩 정도로 수행할 수 있는 음성인식기를 필요로 하지만, 소량 생산하거나, 시제품등에 적용할 경우에는 어쩔 수 없이 DSP나 RISC CPU 기반의 고가의 하드웨어 모듈을 이용하고 있다.

본 논문은 음성인식 칩처럼 저렴하면서도 최소의 부품으로 구성된 음성인식기 구현에 관한 것이다. 음성인식 칩은 음성인식에 필요한 모든 유닛을 포함하고 있다. 즉, 음성인식 유닛 자체뿐만 아니라, A/D 변환기, 메모리, 음성인식기를 제어할 콘트롤러등을 포함하고 있다. 따라서 음성인식 칩을 이용하게 되면 최소한의 부품으로 음성인식 시스템을 구현할 수 있다. 메모리를 내장하고 있기는 하

지만, 내장할 수 있는 용량과 메모리 종류에 제한이 있기 때문에 음성인식 칩을 사용하는 경우에도 외장 메모리를 사용하게 된다. 특히, 화자 종속의 경우는 사용자가 훈련시킨 명령어를 저장해야하기 때문에 플래쉬 메모리가 필수적인데, 현재의 음성인식 칩들은 플래쉬 메모리를 내장하고 있지 않기 때문에 일반적으로 음성인식 칩과 외장 메모리, 두 개의 칩으로 구성된다. 그러나 DSP를 이용하여 하드웨어 모듈로 구현하는 경우에는 DSP, A/D(코덱), 플래쉬 메모리, SRAM과 같이 최소 4 개의 부품으로 구성된다. 경우에 따라서는 DSP에 내장된 SRAM 만으로 충분히 구현 가능하다면 외장 SRAM은 사용되지 않을 수 있다. 반면, 많은 제어 기능을 요구할 경우 별도의 마이크로콘트롤러를 필요로 할 수 있다. 따라서 음성인식 칩과 유사한 형태가 되는데 있어서 가장 중요한 부분은 DSP가 코덱을 내장하고 있는가 하는 점이다. 만약, DSP가 코덱을 내장하고 있다면, 형태적으로는 음성인식 칩과 유사해질 수 있다. 그러나 불행히도 코덱을 내장하고 있는 DSP는 거의 없으며 일부 기존의 코덱을 DSP와 단일 패키지 형태로 만든 DSP가 있기는 하지만 상당히 고가이므로 의미가 없다. 따라서 본 논문에서는 기존의 음성인식기에서 코덱의 역할을 할 수 있는 A/D 변환기를 내장한 저가의 TMS320C2000 계열 DSP를 이용하여 단일 칩 음성인식기를 구현하였다. TMS320C2000 계열 DSP에 내장된 A/D 변환기는 음성인식기에서 주로 사용되는 sigma-delta modulation 방식의 코덱과 달리, 일종의 단순한 sample & hold 회로이기 때문에 anti-aliasing을 위한 저역 필터가 별도로 추가되어야 한다. 한편, TMS320C2000 계열 DSP는 내부에 충분한 용량의 플래쉬 메모리를 내장하고 있기 때문에, 별도의 외장 플래쉬 메모리를 필요로 하지 않는다. 따라서, signal conditioning을 위해 추가된 OP 앰프를 제외하면, DSP 단일 칩으로 음성인식기를 구현할 수 있다.

2. TMS320C2000 계열 DSP를 이용한 화자종속 인식기 구성

TMS320C2000 계열 DSP는 DSP의 신호처리 기능과 범용 마이크로컨트롤러(흔히 마이컴이라고 부른다.)의 제어 기능을 모두 가지고 있기 때문에 Digital Signal Controller(DSC)라고도 불리운다. 과거 DSP들에는 없었던 PWM, UART, A/D 변환기, watchdog 타이머 등의 주변장치를 포함하고 있기 때문에 이런 주변 장치들을 위하여 더 이상 마이크로컨트롤러를 추가적으로 사용할 필요가 없다. 특히 TMS320C2000 계열 DSP는 디지털 모터 제어에 최적의 프로세서로 알려져 있다. 본 논문에서는 TMS320C2000 계열 DSP 중에서 TMS320F2808과 TMS320F2801 DSP를 이용하였다. 이 두 DSP는 내장하고 있는 메모리 용량에 차이가 있을 뿐 pin-to-pin 호환되는 동일한 프로세서이다. 다음은 TMS320F280x DSP의 특징을 간단히 요약한 것이다.

- 16x16 Dual MAC 및 32x32 MAC 연산을 지원하는 32bit CPU
- Atomic 연산 및 매우 빠른 인터럽트 반응 지원
- Harvard Bus 구조 및 통합된 메모리 프로그래밍 모델
- 다양한 용량의 내부 SRAM 및 플래쉬 메모리 내장
- SCI, SPI, CAN, PWM, 12bit A/D변환기, 타이머, GPIO등 다양한 주변장치 내장
- 저가화를 위한 외부 메모리 버스 생략

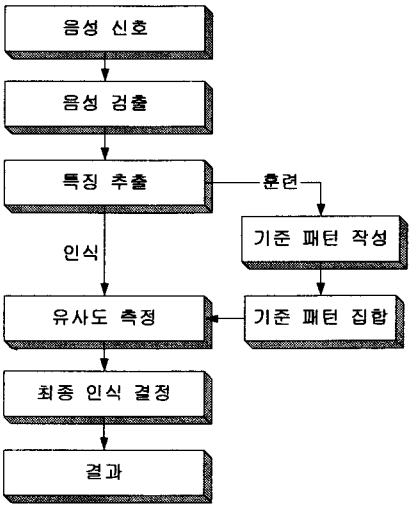

그림. 1 DTW를 이용한 화자 종속 음성인식기

<그림 1>은 본 논문에서 구현한 화자 종속 음성인식기의 블록도이다. 음성 검출과 동시에 특징벡터 추출이 실시간으로 수행된다. 훈련을 위해서는 명령어를 두 번 발음하게 되는데, 두 번 발성된 특징벡터 열은 클러스터링 과정을 거쳐 하나의 기준 패턴의 형태로 플래쉬 메모리에 저장된다. 또한 두 번 발성한 각각의 특징벡터 열을 서로 DTW하여 누적 거리를 구하고 이를 저장해 두었다가 인식 시에 거부를 위한 후처리 단계에서 사용하게 된다. 인식을 수행하는 경우에는 추출된 특징벡터 열을 훈련 시 저장해 놓은 기준 패턴들과 차례로 DTW를 수행하여 인식 결과를 얻은 후, 후처리 과정을 거쳐 인식 결과를 거부할지 아니면 최종 인식결과로 확정할지를 수행하게 된다.

3. 고정 소수점 연산 방식을 이용한 화자 종속 인식기

8 KHz로 표본화된 음성신호는 DC 오프셋을 제거하기 위하여 해당 프레임의 평균을 뺀 후 끝점 검출기로 입력된다. 끝점 검출기는 통상의 방식과 같이 프레임 에너지와 프레임 ZCR을 변형한 LCR(Level Crossing Rate)을 이용하였다[3][4]. 그리고 주변 배경 잡음의 변화에 대응하도록 음성이 아니라고 판명된 프레임의 에너지 및 LCR을 이용하여 이들 파라메터의 문턱값을 적응시켰다. 끝점 검출을 위한 프레임의 길이는 100 샘플(12.5 msec)로 하였다. 한편, 임베디드 음성인식기의 경우 잡음이 심하거나 마이크의 거리가 30 cm 이상 떨어져 사용해야 하는 경우가 있는데 이 경우 ZCR에 기반한 파라메터는 끝점 검출에 큰 도움이 되지 못한다. 따라서 LCR 파라메터의 사용 여부는 옵션으로 설정 가능하게 하였다. 음성으로 판정된 프레임의 음성데이터는 Hamming Windowing 연산과 pre-emphasis 연산이 수행된다. 고정 소수점 연산 오차를 줄이기 위하여 식(1)과 같이 Hamming Windowing 연산과 pre-emphasis를 하나의 연산으로 묶어서 수행하였다.

$$s'(n) = \{s(n) - \alpha s(n-1)\}w(n) = s(n)w(n) - s(n-1)w'(n)$$
$$\text{여기서 } w'(n) = \alpha w(n) \tag{1}$$

식(1)에서 $w(n)$는 Hamming window 함수이며, $s(n)$은 음성 신호이다. α값은 0.95를 사용하였다. $w'(n) = \alpha w(n)$ 연산은 식(1)을 연산하는 과정에서 직접 수행할 수도 있으나 Hamming Windowing 연산과 pre-emphasis 부분이 음성 분석의 초기 부분이므로 고정소수점 연산으로 인한 round-off 오차를 최소화하기 위하여 $\alpha w(n)$을 미리 연산하여 배열로 저장해 두었다. 따라서 $w(n)$과 $w'(n)$배열 두 개의 배열을 사용한다. 그 대신 window 함수의 대칭성을 이용하여 윈도우 값의 절반 씩만을 배열에 저장하여 메모리 사용량을 줄였다.

음성 판정된 프레임에 대하여 특징 추출을 실시간으로 수행한다. 특징 파라메터는 10 차의 LPC cepstrum 파라메터를 사용하였다. 소용량의 화자 종속 인식기에서는 10 차의 LPC cepstrum 파라메터만으로도 충분한 인식률을 얻을 수 있기 때문에 델타 파라메터는 사용하지 않았다. 분석 구간은 25 msec이며 매 프레임 분석 시마다 이전 프레임과 12.5 msec의 오버랩을 하였다. LPC cepstrum 파라메터를 고정소수점 연산을 통하여 계산하는 과정은 자기상관계수, Levinson-Durbin recursion, cepstrum 변환의 세 단계를 거친다. 자기상관계수는 식(2)와 같은 연산을 수행하기 때문에 그 값이 매우 커질 수 있다.

$$r_x(m) = \sum_{n=0}^{N-|m|-1} x(n)x(n+m) \tag{2}$$

따라서 충분한 스케일링을 해주어야 하는데, 고정된 스케일 값을 사용할 경우에는 음성신호가 작을 경우 round-off 오차가 커지게 된다. 이를 해결하기 위하여 매 프레임마다 동적 스케일 값을 사용하였다. 이는 연산의 정밀도를 높일 수 있다는 장점이 있기는 하지만, 이 후 연산에서 정확한 자기상관계수를 얻기 위해서는 각각의 스케일 값을 다시 반영해야한다는 단점이 있다. 그런데, Levinson-Durbin recursion에서는 자기상관계수 값 자체를 사용하는 것 아니라 time-lag에 따른 계수 값의 비를 사용하기 때문에 스케일 값을 반영하지 않아도 된다. 특히 자기상관계수는 특징 벡터 추출에 있어서 가장 초기에 이루어지는 연산이므로 이후 round-off 오차의 영향을 최소화하기 위해서는 가능한 정밀도를 유지할 필요가 있다. Levinson-Durbin recursion과 cepstrum 변환은 고정소수점 연산으로 수행하기에 매우 까다로운 알고리즘들이다. 또한 어느 정도의 연산 정밀도를 유지해야하는지도 고려해야한다. 연산 정밀도는 배정도 연산을 추가하게 되면 정밀도를 원하는 만큼 높일 수 있으나 그 만큼 연산량이 증가하기 때문에 연산량과 연산 정밀도 사이에 적절한 타협이 이루어져야 한다. Levinson-Durbin recursion은 음성 코딩에서 이미 많이 사용되고 있기 때문에 이를 참고하여 정밀도와 연산량을 결정하였다. 특히 G.723.1의 시뮬레이션 코드를 기준 코드로 하여 같은 정도의 연산량을 유지하면서 G.723.1의 시뮬레이션 코드보다 더 정밀한 연산 결과를 얻도록 작성하였다. 고정소수점 연산을 위해서는 LPC 계수의 Q-format을 결정해야하는데, 이는 LPC 계수의 통계적 분포를 고려하여 3.13 Q-format을 사용하였다. 따라서 cepstrum 변환을 통하여 얻어진 LPC cepstrum 계수 역시 3.13 Q-format이 된다. cepstrum 변환은 식(3)을 통하여 얻게 되는데,

$$c(m) = a(m) + \sum_{k=1}^{m-1}\left(\frac{k}{m}\right)c(k)a(m-k) \tag{3}$$

식(3)은 연산 자체가 순환적이며, 나눗셈을 포함하고 있기 때문에 고정 소수점 연산으로 계산하기에 매우 까다롭다. 따라서 부동소수점 연산과의 비교를 통하여 확실한 검증을 통해 고정소수점 연산을 수행하였다.

특징 벡터가 검출되면 식(4)의 DTW를 통하여 인식을 수행한다[5].

$$D = \sum_{n=1}^{N} d(T(n), R(w(n))) \tag{4}$$

이때 $d(T(n), R(w(n))$는 T의 n번째 특징벡터와 R의 $w(n)$번째 특징벡터의 국부적 거리이며, DTW는 <그림 2>와 같이 두 패턴 간의 누적거리 D를 최소화하도록 하는 (n,m) 평면상의 최적경로 $m = w(n)$를 찾는 방법이다.

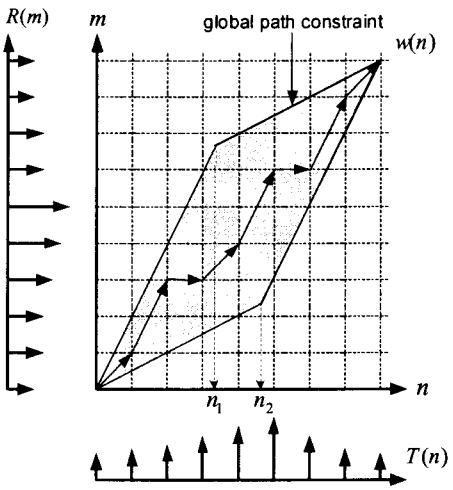

그림 2. DTW를 이용한 두 음성신호 간의 Time Warping

<그림 2>와 같은 전역 경로 제한을 적용하였으며 국부 경로 제한은 Itakura 국부 경로 제한을 사용하였다[6]. Itakura 국부 경로 제한은 다른 국부 경로 제한과 비교하여 인식률은 큰 차이가 없는 반면, DTW시 필요로 하는 메모리를 최소화할 수 있는 장점이 있다. DTW 연산은 <그림 2>에서 x축 한 지점에서 전역 경로 제한안에 포함된 모든 y 지점들에 대한 누적경로를 계산하게 되는데 이때 Itakura 국부 경로 제한을 사용할 경우에는 $x-1$ 지점에서의 모든 y 지점들의 누적값들 만을 기억하고 있으면 되기 때문에 메모리 사용에 제한이 있는 임베디드 시스템에서는 매우 적합한 방식이다. 한편, 고정소수점 연산과 관련해서는 특징분석에서 얻어진 3.13 Q-format의 LPC-cepstrum 벡터 간의 유클리디안 거리를 이용하여 국부적 거리를 구하고, 이를 누적하여 계산하게 되는데 최적 누적

값에 해당하는 변수를 unsigned long형(32bit)으로 선언하면 overflow 없이 연산이 가능하기 때문에 단순히 정수 연산을 통하여 비교적 간단히 계산할 수 있다.

인식 과정에서는 위에서 언급한 DTW를 통하여 최적 누적거리 만을 계산하는 반면, 훈련 과정에서는 최적 누적거리 외에 DTW과정에서 얻어지는 최적 경로를 저장하게 된다. 훈련을 하기 위해서는 해당 명령어를 두 번 발음하게 되는데, 최적 경로는 두 번 발음된 음성을 클러스터링 하는데 이용되며, 최적 누적 거리 값은 후처리 단에서 거부와 관련하여 사용된다. 일반적으로 클러스터링 과정에서는 다수의 패턴으로부터 대표적인 패턴을 얻게 되지만 본 인식기에서는 두 개의 패턴을 사용하므로 식(5)를 통하여 두 개의 특징벡터 열로부터 인식 시에 사용할 하나의 기준 패턴을 얻게 된다.

$$T(n) = \frac{T_1(n) + T_2(w^*(n))}{2} \tag{5}$$

식(5)에서 T_1과 T_2는 훈련 시 두 번 발성을 통하여 얻어진 특징 벡터열이며, $w^*(n)$은 두 벡터 열을 DTW하여 얻는 최적 경로이다.

후처리 단에서는 훈련 과정에서 얻은 DTW 최적 누적 거리 값을 이용하여 인식된 단어를 거부할 지를 판별한다. 만약 인식 결과가 i번째 명령어라면

$$D > kD_i + \beta \tag{6}$$

식(6)이 만족되면 거부하게 된다. D_i는 훈련 시 얻은 i번째 명령어의 최적 누적 값이고 k는 거부율을 조정하는 인자로 이 값을 크게 하면 거부율이 낮아지게 된다. 한편, β는 주변의 소음정도를 판별하여 소음 정도에 비례하는 오프셋 값이다. 주변이 시끄러울 경우 비교적 정확한 발음에 대하여도 거부가 되는 경향이 있는데 이 오프셋 값은 이러한 현상을 완화시켜준다.

4. TMS320F280x DSP를 이용한 하드웨어 구현

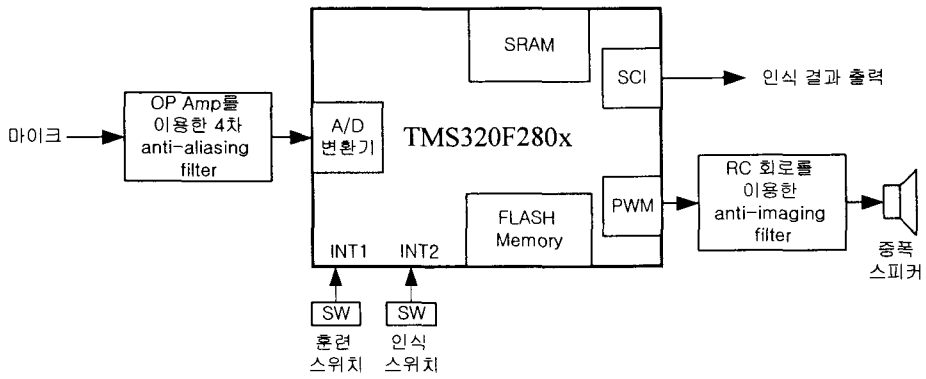

그림 3. TMS320F280x DSP를 이용한 화자 종속 인식기

<그림 3>은 본 논문에서 DSP를 이용하여 구현한 화자 종속 인식기의 하드웨어 구성도 이다. TMS320F2801과 TMS320F2808 두 종류의 DSP를 이용하여 구현하였다. 두 DSP는 내장하고 있는 내부 메모리 용량의 차이를 제외하면 동일한 DSP이다. 다음은 TMS320F280x DSP가 내장하고 있는 메모리의 량을 나타낸다[7].

표 1. TMS320F280x DSP의 내장 메모리

DSP	SRAM(Kbyte)	플래쉬 메모리(Kbyte)	동작주파수(MHz)
TMS320F2808	36	128	100
TMS320F2806	20	64	100
TMS320F2802	12	64	100, 60
TMS320F2801	12	32	100, 60

내장된 메모리 양에 따라 다양한 DSP가 출시되어 있으므로 인식 명령어의 수에 따라 적절한 DSP를 선택할 수 있다. 사용할 수 있는 메모리의 량은 인식 가능한 명령어의 수 및 인식기의 부가적인 기능에 영향을 준다. 따라서 TMS320F2808의 경우는 플래쉬 메모리에 저장할 수 있는 명령어의 수를 늘리거나, 또는 인식 후 인식 결과를 음성으로 출력하는 기능을 추가 할 수 있다. 이번 구현에서는 TMS320F2808 DSP 경우에는 훈련 시 발성된 음성을 ADPCM으로 압축하여 저장해 두었다가 인식 시에 음성으로 출력하는 기능을 추가 하였다. 뿐만 아니라 플래쉬 메모리의 용량이 충분하므로 훈련 시 필요한 안내 메시지도 저장해 두었다가 출력하도록 하였다.

알고리즘을 DSP에 포팅하는 과정에서 중요한 부분은 코드 최적화이다. 임베디드 음성인식기의 경우는 속도 최적화뿐만 아니라 코드 크기 최적화도 신경을 써야 한다. 특히 TMS320F2801의 경우는 내장 SRAM의 크기가 12Kbyte 밖에 되지 않기 때문에 코드의 크기와 데이터 메모리 사용을 최적화하지 않으면, 인식기 구현 자체가 불가능하다. 우선 속도 최적화의 경우는 기본 연산과 관련된 intrinsic 함수들을 모두 어셈블리로 작성하여 사용하였기 때문에 전체적으로는 C 언어를 이용하여 프로그램하였음에도 불구하고 실시간으로 동작시키는데는 큰 어려움이 없었다. 특히, TMS320C2000 계열의 DSP는 32bit x 32bit의 배정도 곱셈기를 지원하며, 이를 적절히 활용하면 속도 저하 없이 연산 정밀도를 높일 수 있기 때문에 음성인식과 같이 연산량이 많으면서도 연산의 정밀도를 요구하는 응용에 상당히 유용한 DSP이다. 다만, DMA를 지원하지 않기 때문에 A/D 변환기로 입력되는 데이터를 모두 인터럽트로 처리해야하며, 이로 인해 발생하는 오버헤드는 어느 정도 감안해야 한다.

메모리 최적화의 경우는 실제 코드 크기가 작게 생성되도록 프로그램하는 것이 중요하지만, 이는 어느 정도 한계가 있기 때문에 적은 메모리 상에서 인식기를 구현할 경우에는 데이터와 코드의 메모리 배치가 더 중요한 역할을 한다. 즉, 변수의 활용도와 실시간 연산에 있어서 해당 함수가 미치는 영향을 분석하여 내장 SRAM에 적재될 데이터와 코드를 결정해야 한다. 내장 SRAM에 적재되는 데이터와 코드는 다음과 같다.

■ 실시간 처리를 위한 ping-pong 버퍼
■ 지역변수를 위한 스택

■ 실시간으로 프레임을 처리할 때 사용되는 임시적 배열들
■ 실시간 분석과 관련된 함수
■ DTW 함수

이외의 상수 테이블이나 실시간 연산에 거의 영향을 주지 않는 연산 관련 함수 및 초기에 한번 수행되는 초기화 함수들은 플래쉬 메모리에서 직접 실행되도록 하였다. 따라서 플래쉬 메모리에는 실행 코드, 상수 데이터(안내 메시지를 위한 음성데이터 포함), 훈련 시 생성된 기준 패턴과 ADPCM 음성데이터(TMS320F2808의 경우)가 저장되게 된다. 기준 패턴을 구성하는 특징벡터의 데이터들은 16bit short형으로 표현 되는데, 메모리 절약을 위하여 8 bit로 양자화하여 저장하였다. 더 낮은 비트율로 양자화하여 저장하는 것이 가능하나 인식률 및 복잡도 등을 고려하여 8bit로 유지하였다. 이렇게 양자화를 하고, 한 명령어의 길이를 최대 2 초로 제한할 경우 32 Kbyte의 플래쉬 메모리를 내장하고 있는 TMS320F2801에서 최대 10 단어까지 인식이 가능하다. 한편, TMS320F2808의 경우는 인식 결과를 음성으로 출력하기 위하여 훈련 시 발성된 음성을 ADPCM으로 압축하여 저장하게 되는데, 이부분이 비교적 많은 량의 메모리를 사용하게 된다. 따라서 128 Kbyte의 플래쉬 메모리를 포함하고 있는 TMS320F2808에서도 약 10 단어를 인식할 수 있고, 추가적으로 남는 플래쉬 메모리는 안내 메시지를 ADPCM으로 압축 저장해 두었다가 훈련 시에 안내를 위한 음성 메시지로 사용한다. 두 DSP의 경우 모두 인식 결과는 DSP의 SCI 포트를 이용하여 직렬 데이터의 형태로 출력된다. TMS320F280x는 GPIO를 위한 충분한 단자들을 가지고 있으므로 응용에 따라서는 병렬 출력을 하는 것도 가능하다. TMS320F2801의 경우는 적은 플래쉬 메모리 양으로 인하여 훈련 시 음성 안내 메시지를 사용할 수 없다. 따라서 훈련 시 SCI 포트를 통하여 안내 메시지를 문자로 출력한다. 훈련 시 PC의 하이퍼터이널과 같은 프로그램을 이용하면, 훈련과 관련된 안내 메시지의 도움을 받을 수 있다.

<그림 3>의 OP 앰프는 anti-aliasing 및 마이크로폰 신호의 증폭을 수행한다. 사용된 OPA2348 소자는 두 개의 OP 앰프를 내장하고 있다. 이 두 개의 OP앰프를 이용하여 다음과 같은 4-pole 체비세프 저역 필터를 설계하였다.

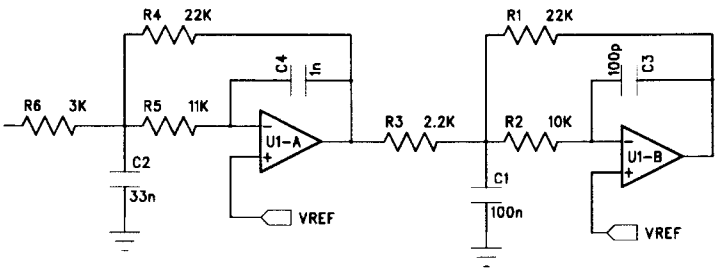

그림 4. OP 앰프를 이용한 4-pole 체비세프 저역 필터

cutoff 주파수는 3.4 KHz이며, 증폭도는 73.3이다. 패스밴드 대역의 특성이 좋은 버터워스 필터는 4차로 구현할 경우, 저지대역 감쇠가 충분하지 못하여 체비세프 필터를 선택하였다. 한편, TMS320F280x

DSP는 D/A 변환기는 내장하고 있지 않기 때문에 TMS320F2808의 경우, 음성 출력을 위하여 내장 주변장치인 PWM을 이용하였다. 부품을 최소화하기 위하여 PWM 출력단의 anti-imaging 저역 필터로는 단순히 RC 회로를 이용하였는데 8 KHz의 음성데이터를 곧 바로 출력할 경우 image에 의한 왜곡이 발생하게 된다. image들의 영향을 최소화하기 위하여 48 KHz로 샘플링 주파수를 변환하는 루틴을 추가하여 up-sampling된 음성을 출력하도록 하였다. <그림 5>는 제작된 음성인식 하드웨어이다. TMS320F280x는 TQFP 및 BGA 타입의 두 종류의 패키지가 있는데, 두 종류 모두를 이용하여 제작해 보았다. 제작의 용이함을 위하여 DSP 및 전원회로를 포함한 상용 모듈을 이용하여 제작하였다.

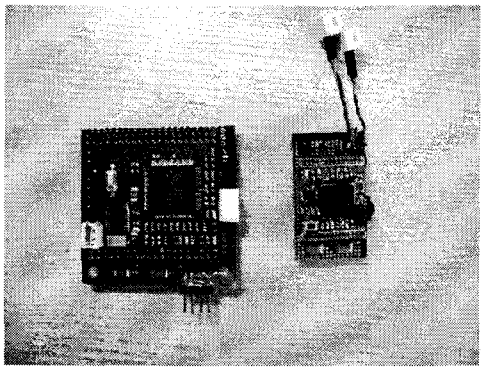

그림 5. 제작된 음성인식 하드웨어

5. 결 론

본 논문에서는 음성인식 칩을 대신할 수 있는 DSP를 이용한 단일칩 화자종속 음성인식기를 구현하는 방법을 제시하였다. 이를 위하여 DTW 기반의 화자 종속 음성인식 알고리즘을 리소스가 매우 제한되어 있는 임베디드 환경에 적합하도록 속도 최적화와 코드 크기 최적화를 수행하여 12Kbyte의 SRAM과 32Kbyte의 플래쉬 메모리만을 내장하고 있는 저가의 TMS320F2801 DSP에서 동작하도록 하였다. A/D 변환을 위하여 TMS320F2801에 내장된 12 bit A/D 변환기를 활용하였으며, anti-aliasing 및 마이크로폰 입력 신호 증폭을위하여 OP 앰프를 추가하여 4차의 체비세프 필터를 설계하여 사용하였다. 또한, 보다 많은 내장 메모리를 포함하고 있는 TMS320F2808을 이용한 경우에는 인식 후 음성 출력 기능을 추가하였다. 이를 위하여 훈련 시, 화자의 음성을 ADPCM으로 압축 저장하여 인식 후 결과를 음성으로 출력하는데 사용하였다. 음성 출력을 위하여 별도의 D/A 변환기를 사용하지 않고 내장된 PWM 주변장치를 이용하여 부품의 추가를 최소화 하였다.

TMS320F2801 DSP를 이용하고 한 명령어의 길이를 2 초로 제한할 경우, 10 단어까지 인식이 가능하였다. 현재 TMS320F2801 DSP의 가격은 1000 개 기준으로 대략 $3 대 수준이고 OP 앰프를 제외하고는 거의 추가의 부품이 필요 없기 때문에 매우 저렴하게 제조할 수 있다. 비록, 음성인식 칩이 가격적인 측면에서는 더 저렴하지만, 일반적으로 소량 구매하여 사용하기 어렵기 때문에, 본 논문에서 구현한 저렴한 가격의 DSP를 이용한 단일 칩 음성 인식기의 활용이 기대된다.

참 고 문 헌

[1] 한국전자통신연구원 2001. 음성처리 시스템 기술/시장 보고서.
[2] Junqua, Jean-Claude. 2000. *Robust Speech Recognition in Embedded Systems and PC Applications.* Dortrecht: Kluwer Academic Publishers.
[3] Rabiner, L. & Schafer, R. 1978. *Digital Processing of Speech Signals.* New York: Prentice Hall.
[4] Rabiner, L. & Juang, B. 1993, *Fundamentals of speech recognition.* New York: Prentice Hall.
[5] Sakoe, H. & Chiba, S. 1978. "Dynamic programmng algorithm optimization for spoken word recognition," IEEE Trans. of Acoustics, Speech and Signal Processing. Vol. ASSP-26, No.1, 43-49.
[6] Itakura, F. 1978. "Minimum precition residual applied speech recognition," IEEE Trans. of Acoustics, Speech and Signal Processing Vol. ASSP-26, No.6, 575-582.
[7] Texas Instruments. 2006. *TMS320F280x Digital Signal Processors Data Manual.*

접수일자: 2007. 10. 12
게재결정: 2007. 11. 30

▲ 정익주
강원도 춘천시 효자2동 (우: 200-701)
강원대학교 전기전자 공학부
Tel: +82-33-250-6322
E-mail: ijchung@kangwon.ac.kr