

---

# 소어휘 단어단위의 음성인식 칩 설계

안점영\* · 최영식\*

The Design of Speech Recognition Chip for a Small Vocabulary as a Word-level

Jeom-Young Ahn\* · Young-Shig Choi\*

---

이 논문은 2000년도 동의대학교 학술연구비 지원으로 수행되었음.

---

## 요 약

소어휘 단어단위의 음성을 인식할 수 있는 음성인식 칩을 설계하였다. 설계된 칩은 음성 신호의 시작과 끝점 검출 부분, LPC 캡스트림 계수 추출 부분, DTW 실행 부분과 외부 메모리 인터페이스 부분으로 구성되어 있다. CMOS 0.35um TLM 공정으로 설계된 이 칩은 4x4mm<sup>2</sup>의 면적에 126,938개의 게이트로 만들어져 있다. 그리고 전용 H/W의 동작 속도는 5MHz에서 60MHz까지 조정 가능하다. 5MHz 클럭을 사용하는 경우, 50~60 프레임 정도의 소어휘 단어 단위의 음성을 초당 100,000개까지 비교할 수 있는 능력이 있고, 60MHz의 클럭을 사용하는 경우는 초당 1,200,000개의 단어를 비교할 수 있다.

## ABSTRACT

A speech recognition chip that can recognize a small vocabulary as a word-level has been designed. It is composed of EPD(Start and End-point detection) block, LPC block, DTW block and external memory interface block. It is made of 126,938 gates on 4x4mm<sup>2</sup> area with a CMOS 0.35um TLM process. The speed of the chip varies from 5MHz to 60MHz because of its specific hardware designed for the purpose. It can compare 100,000 voices as a small vocabulary which has approximately 50~60 frames at the clock of 5MHz and also up to 1,200,000 voices at the clock of 60MHz.

## 키워드

음성인식, LPC 캡스트림 계수, DTW, 칩 설계

## 1. 서론

고도로 발달된 정보통신 환경에서 사람과 기계간의 원활한 정보교환을 위해 음성신호처리 기술이 꾸준히 연구되고 있다. 그 중에서 음성인식기술은 사람의 언어를 해석하여 그에 따른 적절한 행동을 취할 수 있는 기계를 만드는 것으로서 정보산업, 디지털 통신, 가전 분야 및 멀티미디어 등에 지대한 파급효과가 있으며, 현재 PCS, 셀룰라폰, 자동차, 증권정보, 안내시스템 등에 광범위하게 활용되고 있다. 이들 음성인식 시스템은 범용 DSP나 전용 프로세서로 구현된다<sup>[1]-[4]</sup>.

본 연구는 소어휘 단어단위의 음성인식 칩 개발에 최종 목표를 두고 있다. 따라서 처리 데이터들을 가능한 최소화하고, 계산량을 줄여 칩의 크기를 작게 하는데 주안점을 두고 설계하였다. 음성인식 알고리즘은 DTW (Dynamic Time Warping)를 이용하였다<sup>[5]</sup>. 이 알고리즘은 워핑(Warping) 함수를 찾는 과정에서 많은 계산이 필요하므로 기준음성의 개수가 많아지면 실시간 처리가 어려워지게 된다. 그러므로 DTW 알고리즘은 주로 소용량 어휘의 고립단어 인식에 이용된다. DTW는 인식시간이 많이 소요된다는 단점이 있지만, 인식률이 높아서 VLSI칩으로 구현되어 현재 많이 상용화되어 있다. 본 논문에서 설계한 전용 H/W의 동작 속도는 5MHz에서 최대 60MHz까지 조정 가능하다. 그러므로 동작속도가 60MHz인 경우, 50~60 프레임 정도의 소어휘 단어단위의 음성이라면 매초 1,200,000 개까지 비교할 수 있는 매우 빠른 음성처리 능력을 갖는다.

본 논문의 구성은 다음과 같다. II장에서는 인식시스템의 구성과 인식 알고리즘에 대해 간략히 기술하고, III장에서는 이 알고리즘을 실제 칩으로 구현하기 위한 과정과 주요 블록을 상세히 설명하고 나아가 주요 블록의 레이아웃 및 시뮬레이션 결과에 대해 설명한다. 마지막으로 IV장에서는 결론 및 추후 연구방향에 대해서 설명한다.

## II. 인식 알고리즘

마이크를 통해 들어온 음성신호는 3.5KHz의 아날로그 저역통과필터를 거쳐 ADC에서 8KHz로 샘플링

되고, 8bit로 양자화 되어 디지털신호로 변환된다. 변환된 음성 데이터의 고주파 부분을 강조하기 위해  $1 - 0.95z^{-1}$ 의 고역통과 디지털 필터로 프리엠퍼시스 한다.

그리고 각 프레임마다 영교차율(ZCR)과 평균자승 에너지를 계산하여 음성신호의 시작과 끝점을 검출한다. 영교차율과 평균자승 에너지 계산은 식(1), (2)와 같다.

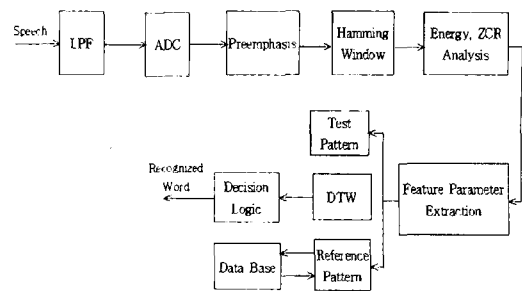


그림 1. 음성인식 시스템  
Fig. 1 Speech Recognition System

$$Z(m) = \frac{1}{N} \sum_{n=m-N+1}^m \frac{(|\text{sgn}(s(n))| - |\text{sgn}(s(n-1))|) + 1}{2} u(m-n) \quad (1)$$

$$\text{sgn}(s(n)) = \begin{cases} +1, & s(n) \geq 0 \\ -1, & s(n) < 0 \end{cases}$$

$$E(m) = \sum_{n=m-N+1}^m \{s(n)w(m-n)\}^2 \quad (2)$$

여기서 N은 한 프레임의 음성 샘플수이고,  $s(n)$ 은 프레임 내부 n번째의 음성샘플,  $w(m-n)$ 은 해밍창 함수이다. 시작과 끝점 검출이 끝나면 해밍창(100byte 길이)을 25byte씩 이동하면서 자기상관계수를 구하고, 이 계수가 다시 Durbin 방법에 의해 LPC(Linear Predictive Coding) 계수로 변환된다. 이에 관련된 수식들은 다음과 같다.

$$E^{(0)} = R(0) \quad (3)$$

$$k_i = [R(i) - \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j)] / E^{(i-1)}, \quad 1 \leq i \leq p \quad (4)$$

$$\alpha_i^{(i)} = k_i \quad (5)$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}, \quad 1 \leq j \leq i-1 \quad (6)$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)} \quad (7)$$

여기서  $p$ 는 LPC 예측차수,  $\alpha_m^{(p)}$ 는 LPC 계수,  $k_m$ 은 편자기상관계수,  $R(i)$ 는 자기상관계수, 그리고  $E^{(i)}$ 는 에너지이다. 켈스트럼으로 불리어지는 단구간 대수진폭 스펙트럼의 역푸리에 변환은 음성인식의 경우 LPC 계수보다 더욱 신뢰할 수 있는 특징으로 알려져 있다[6]. 따라서 LPC 계수를 다음과 같이 12차의 LPC 켈스트럼 계수로 변환한다.

$$c_0 = \ln \sigma^2 \quad (8)$$

$$c_m = \begin{cases} a_m + \sum_{k=1}^m \left(\frac{k}{m}\right) c_k a_{m-k}, & 1 \leq m \leq p \\ \sum_{k=1}^m \left(\frac{k}{m}\right) c_k a_{m-k}, & m > p \end{cases} \quad (9)$$

여기서  $\sigma^2$ 은 LPC 모델의 이득,  $c_m$ 은 켈스트럼 계수이다.

지금까지 각 프레임별로 분석된 특징벡터 열(sequence)을 가지고 각 어휘의 표준패턴을 만들어 메모리에 저장한다.

화자는 인식대상 어휘를 수회씩 발성하고, 발성된 음성마다 표준패턴이 만들어지므로 표준패턴의 수는 발성된 음성 수만큼 많다. 마이크에서 인식을 위한 미지의 음성신호가 들어오면 표준패턴 작성과 같은 동일한 절차에 따라 시험패턴이 만들어진다. 시험패턴과 모든 표준패턴을 비교하여 유사도가 가장 큰 표준패턴을 찾아 이것을 미지의 음성으로 인식한다.

동일한 음성이라도 화자 또는 발성환경에 따라 음성의 지속시간이 변하므로 음성의 시계열 패턴도 달라져서 서로 다른 음성으로 보인다. 본 음성인식 시스템에서는 이와 같이 음성의 지속시간의 차이로 인한 영향을 제거하기 위하여 음성신호 내부의 각 부분을 늘이거나 줄이는 방법으로 시간축을 정렬하여 비교하는 DTW를 이용한다. DTW는 시험패턴과 표준패턴의 두 특징벡

터들 사이의 전체 누적 거리가 최소인 경로를 워핑함수(Warping Function)로 구한다. 최적 워핑경로를 얻기 위해 사용되는 전체 거리 측정은 식(10)과 같이 정의된다.

$$D[i(k), j(k)] = \frac{\sum_{k=1}^K d[i(k), j(k)] W(k)}{N(w)} \quad (10)$$

여기서  $D[i(k), j(k)]$ 는 경로의 길이  $K$ 에 따른 전체 거리로 주어진 함수이고,  $d[i(k), j(k)]$ 는 시험패턴의 프레임  $i(k)$ 와 표준패턴의 프레임  $j(k)$ 의 소구간 거리이며,  $W(k)$ 는  $k$ 번째 소구간 경로의 웨이팅 함수이고,  $N(w)$ 는 웨이팅 함수  $W$ 의 함수로서 정규화 인자이다.

최적경로는 전체거리  $D[i(k), j(k)]$ 의 최소값으로 정의되고, 위의 식(10)에서 직접 구해진다.

$$D_T = \min_{(K, i(k), j(k))} [D(i(k), j(k))] \quad (11)$$

시험패턴과 각 표준패턴간의 최적경로를 측정 한 후 이들 중 가장 작은 크기의 최적경로를 갖는 표준패턴이 인식음성으로 결정된다.

### III. VLSI 구현

#### 3.1 설계 순서

그림 2는 알고리즘 개발에서부터 최종 레이아웃까지의 설계 순서를 나타내고 있다. 이 음성 인식 칩은 크게 EPD, LPC, DTW 및 MEM\_IF 블록으로 구성되어 있다. 칩의 전반부 설계인 알고리즘 개발은 C 언어를 사용하였다.

C 언어로 최적의 알고리즘을 개발한 후 하드웨어로 구현하기 위하여 회로합성이 가능한 HDL인 Verilog 언어의 형태로 변환하였다. 회로 합성 후 전단계 시뮬레이션을 진행하여 칩이 제대로 동작하는지 확인하였으며, 배치 및 연결을 하기 전에 Floorplanning을 진행하여 칩의 크기를 줄일 수 있도록 하였다. 마지막으로

레이아웃이 제대로 되어있는지를 확인한 다음 후단계 시뮬레이션을 수행하여 칩 설계를 완료하였다.

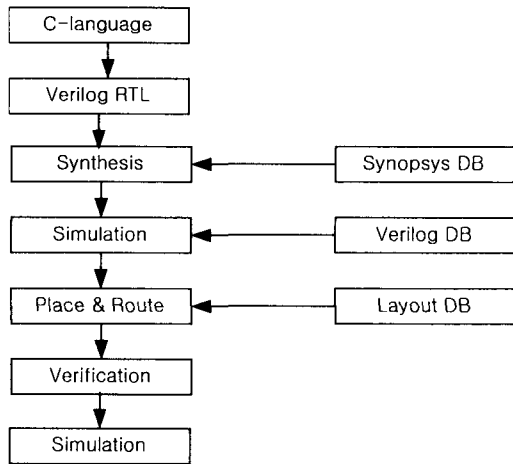


그림 2. 설계 순서  
Fig. 2 Design Flow

### 3.2 전체 구성도

음성인식 알고리즘을 칩으로 구현하기 위해 4개의 중요 블록으로 나누었다. 그림 1에서 아날로그 블록인 저역 통과 필터 LPF 및 Analog-to-Digital Converter(ADC)는 그림 3의 전체 구성도에서 제외되어 있다. 본 논문에서는 디지털 부분만 칩으로 구현하는 것을 목표로 하였기 때문이다. 그러나 실제 제품으로 만들기 위해서는 LPF 및 ADC 블록이 함께 만들어져야 한다. 그림3을 보면 이 칩은 입력 신호의 시작과 끝점을 검출하는 EPD(End-point detection) 블록, LPC 캡스트림 계수를 구하는 LPC 블록, 패턴의 유사도를 측정하는 DTW 블록과 외부 메모리를 효율적으로 연결해주는 MEM\_IF(Memory interface) 블록으로 구성되어 있다. 전체 구성도의 외부 연결 I/O는 모두 88개인데, 그중에서 외부 메모리를 연결해 주는 I/O가 가장 많고 나머지는 입력, 출력 그리고 VDD와 VSS를 연결해주는 I/O들이다. 여기서 입력 I/O는 모두 13개이며 이중 8개는 입력 신호를 받아들이고 나머지는 Reset, Frame start enable, Clock 등이다. 출력 I/O는 13개이고, VDD와 VSS는 각각 12개로 구성되어 있으며 이중 VSS I/O는 8개, VDD I/O는 4개이다. VSS I/O는 전체 레이아웃을 할 때 ground bouncing을 고

려하여 VDD I/O보다 개수를 많이 배치하였다. CMOS 0.35um TLM 공정 라이브러리로 합성한 결과 전체 게이트 수는 126,938개이며 5MHz에서 60MHz까지 광범위하게 동작한다.

### 3.3 EPD 블록

음성인식 칩을 실제로 구현하기 위해서는 음성신호의 시작과 끝점을 정확히 찾아내는 것이 아주 중요하다. 시작과 끝점은 음성인식의 정확도에 영향을 주기 때문이다. 여기서는 각 프레임의 영교차율(ZCR)과 평균자승 에너지를 계산하여 음성신호의 시작과 끝점을 검출한다.

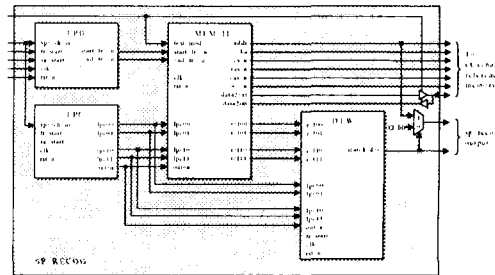


그림 3. 칩 전체 구성도  
Fig. 3 Architecture of the chip

주 영교차(Zero Crossing)에 의한 ID(Increment /Decrement)와 에너지 계산이라는 두 개의 간단한 측정값에 기반을 두고 동작이 수행된다. 이에 앞서 입력 신호는 음성내의 본질적인 고주파 영역의 보정을 위해 pre emphasis를 거치게 된다. EPD 블록의 구조는 그림4에 나타나 있다. pre\_amp 블록에서 고주파 영역이 보정된 신호가 zcross 블록과 erg\_clac 블록으로 입력된다. zcross 블록에서는 연속된 세 개의 입력 신호를 이용하여 영교차를 찾아내고, erg\_clac 블록에서는 해당 프레임의 에너지를 계산한다. 이 두 블록의 결과를 이용하여 다음 단의 두 start\_point 블록과 end\_point 블록에서 신호의 시작점과 끝점이 구해진다.

### 3.4 LPC 블록

시작과 끝점을 검출한 후 음성 샘플은 LPC 블록으로 인가되어서 자기상관계수를 생성한다. 이렇게 생성

된 자기상관계수는 Durbin방법에 의해 LPC 계수로 전환되고, 이 계수로부터 음성인식에 유용한 LPC 캡스트럼 계수가 도출된다.

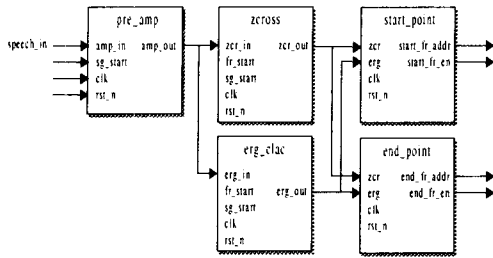
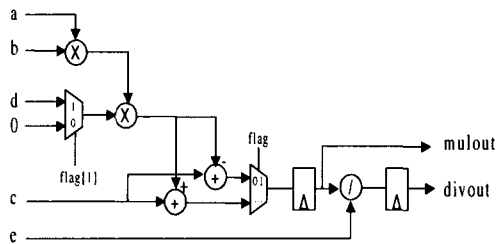


그림 4. EPD 구조  
Fig. 4 Architecture of EPD

LPC 캡스트럼 계수를 도출해내기 위해서는, 식(3)에서 식(9)까지 연산을 반복 수행하여야 한다. 일반적으로 각 식마다 연산을 위해 하드웨어가 한 개씩 필요하지만 본 논문에서는 이러한 연산들을 그림 5와 같은 하나의 데이터 처리경로에 의해 공유·수행되도록 설계하여 하드웨어의 이득을 꾀하였다. 이러한 연산 순서는 FSM(Finite State Machine)으로 만들어진 제어 신호에 의해 반복·실행된다.



flag	2'b00	2'b01	others
action	(a*b)+c	c-(a*b)	(a*b*d)+c

그림 5. LPC 블록 구조도 및 동작 상황  
Fig. 5 Architecture of LPC and working status

### 3.5 DTW 블록

음성 인식 시스템에서 가장 문제가 되는 것 중 하나는 많은 양의 표준패턴을 빠른 속도로 시험패턴과 비

교하는 것이다. 이를 위해 많은 시도가 있었으며, 특히 구현을 용이하게 하기 위해 유연성(flexibility)과 모듈 방식(modularity)이 고려되었다.

본 논문에서 표준패턴과 시험패턴의 비교를 위해 사용된 DTW 알고리즘은 표준패턴과 미지의 시험패턴 간의 비선형 패턴 매칭을 수행하는 고갈 탐색 기술(exhaustive search technique)

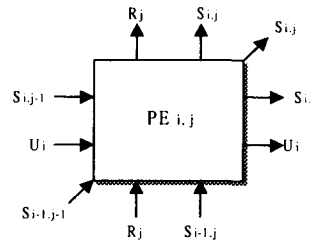


그림 6. 처리 요소  
Fig. 6. Process element

알고리즘이다. 처리 요소(PE : Processing Element)는 그림 6과 같고, 각 PE는 systolic array 구조로 이웃하는 6개의 PE와 연결된다. 처리요소는 좌측  $S_{i-1,j}$ 에서 하단  $S_{i,j}$ 까지 5단자로부터 입력을 받아 상단  $R_j$ 에서 우측  $U_i$ 까지 5단자로 출력신호를 내보낸다. (i, j)번째 PE의 출력값( $S_{i,j}$ )은 미지의 시험패턴 중 i번째 프레임 신호  $U_i$ , 표준패턴 중 j번째 프레임 신호  $R_j$ , 그리고 이웃하는 PE의 출력들( $S_{i-1,j-1}$ ,  $S_{i-1,j}$ ,  $S_{i,j-1}$ )로부터 다음과 같이 결정된다<sup>[7][8]</sup>.

$$S_{i,j} = \min[ S_{i-1,j}+D_{i,j}, S_{i-1,j-1}+2*D_{i,j}, S_{i,j-1}+D_{i,j} ] \quad (12)$$

여기에서  $D_{i,j}$ 는  $R_j-U_i$ 의 절대값을 나타낸다. 위 식을 다시 쓰면 다음과 같다.

$$S_{i,j} = \min[ \min[ S_{i-1,j}+D_{i,j}, S_{i-1,j-1}+D_{i,j} ], S_{i-1,j-1}+2*D_{i,j} ] \\ = \min[ \min[ S_{i-1,j}+|R_j-U_i|, S_{i-1,j-1}+|R_j-U_i| ], S_{i-1,j-1}+\text{shift\_left1}(|R_j-U_i|) ] \quad (13)$$

그림 7은 systolic array 구조로 구성된 DTW 블록으로서 좌측 최하단의 PE에서부터 우측 최상단의 PE

까지 경로에 따른 최소거리를 식(12)에 의해 최적의 누적 거리를 산출하여 정합 인자를 계산한다. 이러한 구조는 일종의 23단 파이프라인 형태로써 23 클록의 초기 지연 경과 후에는 매 클록마다 출력값을 산출하게 된다. 5MHz 클록을 사용하는 경우, 50~60 프레임 정도의 고립단어를 초당 100,000개까지 비교할 수 있는 처리 능력(초기 지연 시간 0.23msec)을 가지며, 60MHz의 클록을 사용하는 경우는 초당 1,200,000개의 고립단어를 비교할 수 있다(초기 지연 시간 19.15 $\mu$  sec, 물론 메모리 데이터 액세스를 위한 ROW active, RAM data delay 등을 고려하면 위 수치보다 약간 낮아질 수 있다). 이는 Oxford Compact 영영 사전의 수록 단어 수가 총 187,000 단어인 것을 고려할 때 매우 빠른 처리 능력임을 알 수 있다.

DTW 알고리즘은 음성의 음소 길이가 부분적

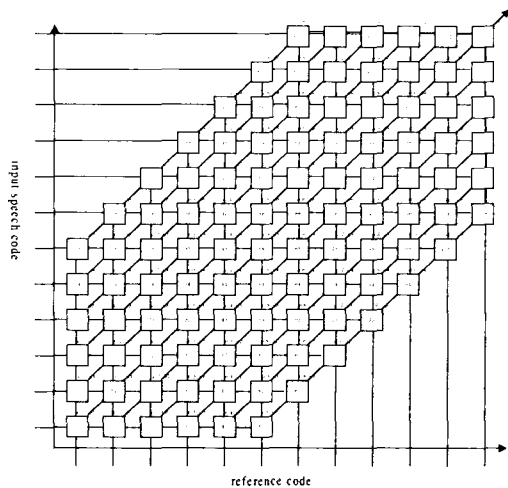


그림 7. DTW의 Systolic 배열 구조  
Fig. 7 Systolic array of DTW

으로 길게 또는 짧게 받음되는 경우에도 그 비교가 가능하며, 기본 원리를 다음 그림 8에 제시하였다. 그림 8(a)는 시험패턴과 표준패턴이 완전히 동일한 예이고, 그림 8(b)는 두 패턴의 전체 지속시간은 같지만 부분적으로 그 음소의 길이가 다른 경우이다. 이해를 돕기 위해 한 프레임의 12개 LPC 캡스트럼 계수를, “바둑이”라는 단어의 음소로 대신하였다. Tilde(~)는 그전의 음소가 길게 받음된 것을 나타낸다. 두 경우 모두 동일

한 패턴으로 인식한다. 이러한 테스트 벡터에 의한 시뮬레이션 결과는 3.8절에 제시되어 있다.

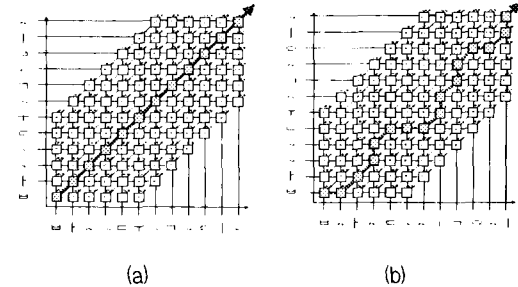


그림 8. DTW의 기본 동작 원리  
Fig. 8 Basic operating examples of DTW

### 3.6 메모리 인터페이스와 신호 제어 블록

외부 메모리에 저장되어 있는 표준패턴 신호에 효율적으로 접근하기 위해 완전히 H/W로 구성된 메모리 인터페이스 블록을 설계하였다. 이 메모리 인터페이스 블록은 14개의 Address I/O를 가지고 있으며 한번에 32bit 데이터를 불러들일 수 있다. 그림 3의 전체 칩 구성도에는 표시되어 있지 않지만 신호 제어 블록은 입력신호가 칩으로 들어와 정확한 경로를 통해 처리되어 출력될 수 있도록 전체 칩을 제어하는 신호를 생성한다.

### 3.7 회로 합성

시뮬레이션을 통해 검증된 RTL 코드는 Synopsys사의 합성 툴을 이용하여 블록 별로 게이트 레벨 넷리스트(Gate-level netlist)로 합성하였다. 광범위한 동작 속도를 가질 수 있도록 합성하였고, 넓은 범위의 동작 속도에서 면적을 최소화하는데 중점을 두었다. 이러한 음성인식 칩이 다양한 환경, 특히 가장 나쁜 조건인 Vdd=3.0V, Temp=70 $^{\circ}$ C에서 게이트 수를 늘리지 않고 5MHz~60MHz까지 동작하도록 합성하는데 많은 어려움이 있었다.

기존의 방식으로 합성 라이브러리를 구축한 결과 최악 조건에서 지연인자(Derating factor)가 지나치게 크게 설정되고, Interconnect 커패시턴스가 게이트 커패시턴스와 거의 같은 정도로 크게 모델링이 되어 게이트당 지연시간이 비현실적으로 크게 되어 도저히 요구되는 속도를 만족시킬 수 없었다. 그래서 라이브러리의 회로합성 조건을 표 1과 같이 수정하였으며, 따라

서 만족스러운 결과를 얻을 수 있었다. 수정된 회로합성 조건에서 각 구성 블록별 크기는 다음과 같다.

- EPD : 1,940 gates
- LPC : 14,778 gates
- DTW : 106,120 gates
- MEM\_IF : 3,584 gates

전체 음성인식 칩의 게이트 수는 126,938개이며 이는 NAND 게이트 수로 환산된 것이다.

표 1. 최악 조건에서의 회로 합성 매개 변수  
Table 1. Synthesis parameters at worst condition

	Before Change	After Change
Process Variation	1.4	1.3
Wire Capacitance	1	0.25
Max Transition	20.0ns	2.0ns
Max Capacitance	345 fF(normal size)	deleted

### 3.8 게이트 레벨 시뮬레이션

합성 완료된 후 블록별로 동작 및 타이밍을 검증하는 시뮬레이션을 수행하였다. 이 경우 각 게이트의 고유 지연 시간 및 팬 아웃과 Routing wire에 의한 Interconnect delay를 모두 고려하여야 한다. Cell 라이브러리에 포함되어 있는 지연 시간들과 합성된 넷리스트로부터 각 네트에 걸리는 지연인자를 계산하고 이를 Standard delay format(SDF)으로 저장한다. 이렇게 만들어진 SDF 파일을 포함시켜서 Verilog 시뮬레이션을 수행하면 최악, 표준, 최상 조건에서의 동작을 정확히 예측할 수 있다. 결과적으로 시뮬레이션에서 나타난 Critical path delay는 합성에서 예측되었던 것과 거의 일치하는 값을 나타내었다. 그림 9와 10은 시뮬레이션 결과를 보여 주고 있다.

그림 9는 EPD 블록의 시뮬레이션 결과로서 음성신호의 시작과 끝점을 지시하는 신호가 나타남을 확인할 수 있다. 그림 10은 DTW 블록의 시뮬레이션 결과로서 시험패턴과 여러 표준패턴 사이의 매칭 결과를 보이고 있다. 전술한 바와 같이 Dynamic Time의 영향을 확인하기 위해 부분적으로 LPC 계수를 시간축에 대해 늘려서 시뮬레이션 해본 결과 예상된 바와 같이 동일한 음성으로 인식, 출력됨을 확인할 수 있었다.

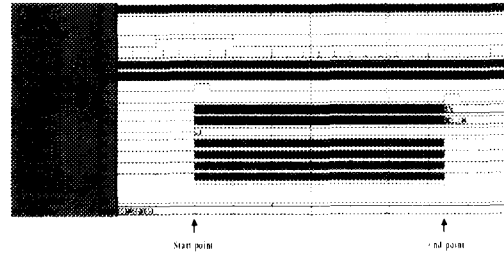


그림 9. EPD 블록의 시뮬레이션 결과  
Fig. 9. Simulation Result of EPD Block

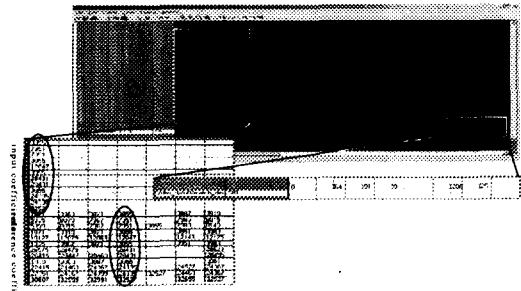


그림 10. DTW 블록의 시뮬레이션 결과  
Fig. 10 Simulation Result of DTW Block

### 3.9. 레이아웃

이 음성인식 칩은 게이트 수가 많지 않기 때문에 블록마다 개별적으로 P&R(Placement & Routing)을 수행하지 않고, 칩 전체에 일괄적으로 적용하는 Flatten 방법으로 P&R을 수행하였다. 이 음성인식 칩은 크기가 작기 때문에 클록 트리 간의 스큐 문제가 발생할 확률도 크지 않고, 칩의 크기가 클 때 발생하기 쉬운 게이트간의 긴 경로 문제도 거의 발생하지 않았다. 레이아웃이 완성된 후 가장 큰 지연시간 값을 갖는 경로를 찾아내어 60MHz 동작에 전혀 문제가 없음을 확인할 수 있었다.

사용된 공정 기술은 CMOS 0.35um TLM이며 레이아웃은 Apollo라는 P&R 툴을 사용하였다. 칩의 크기를 작게 하기 위하여 Floorplanning을 여러 번 시행한 후 진행된 Flatten 방식의 P&R은 126,938개의 게이트를 4x4mm<sup>2</sup>의 크기로 만들었다. 그림 11은 칩 전체의 레이아웃을 보인 것이다.

참 고 문 헌

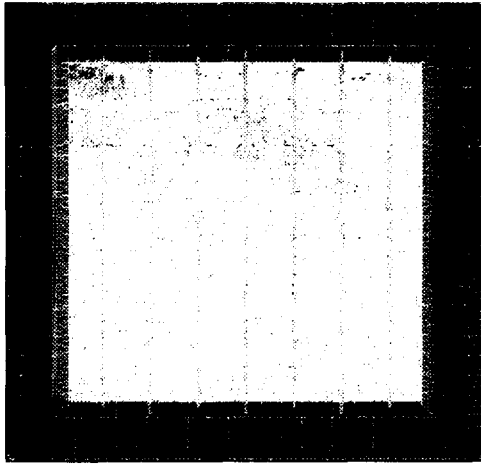


그림 11. 칩 레이아웃  
Fig. 11 Chip layout

IV. 결 론

본 논문에서는 소어휘 단어단위의 음성인식 칩을 전용 H/W로 설계하였으며 음성인식 알고리즘은 DTW를 이용하였다. 칩의 크기를 고려하여 LPC 블록을 하드웨어 공유 구조로 설계하였고, DTW에서 인식 시간이 많이 소요된다는 단점을 보완하기 위해 DTW 블록을 파이프라인 구조로 설계하였다.

사용된 공정기술은 CMOS 0.35um TLM이며 설계된 칩의 크기는 4x4mm<sup>2</sup>이다. 빠른 표준 메모리 인터페이스 블록이 칩에 내장되어 있으므로 메모리에 있는 데이터를 빠른 속도로 불러올 수 있다. 그리고 또한 이 칩은 동작 속도가 5MHz에서 60MHz까지 조정될 수 있으며, 5MHz 클럭을 사용하는 경우 대략 50~60 프레임의 가진 고립 단어를 초당 100,000개까지 비교할 수 있는 능력을 가지며, 60MHz의 클럭을 사용하는 경우는 초당 1,200,000개의 고립단어를 비교할 수 있는 매우 빠른 음성 처리 능력을 가질 수 있다.

앞으로 인식을 향상과 더불어 칩의 크기를 더욱 축소시켜 시장성이 있는 제품을 설계 제작할 계획이다.

[1] Teh C C, Jong C C and Siek L, "Low-cost speech recognition system for small vocabulary and speaker independent" Proceedings of SPIE Design, and Simulation in Microelectronics, pp. 208-211, Nov. 2000

[2] 윤대희 외 5인, "TMS320C31을 이용한 차량항법 음성인식 시스템의 실시간 구현", TELECOM · COMMUNICATIONS REVIEW, Vol. 9, No. 6, pp.984-991, 1999

[3] Soo-Won Kim, et al., " A VLSI Chip for Isolated Speech Recognition System", IEEE Tr. on Consumer Electronics, Vol. 42, No. 3, pp. 458-468, Aug. 1996

[4] An-Nan Suen, Jhing-Fa Wang, and Yuen Lin Chiang, "A Cepstrum Chip : Architecture and Implementation," IEEE International Sym. on Circuit and Systems, Vol. 2, pp. 1428-1432, 1995.

[5] Cody Myers, Lawrence R. Labiner, Arron E. Rosenberg, "Performance Tradeoff in Dynamic Time Warping Algorithms for Isolated Word Recognition," IEEE Tr. on Acoustics, Speech and Signal Processing, ASSP-28, No. 6, Dec. 1980.

[6] Jia-Ching Wang, Jhing Fa Wang, and Yu-Sheng Weng, "Chip Design of MEL Frequency Cepstral Coefficients for Speech Recognition", Proceedings of the 2000 IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 6, pp. 3658-3661, July, 2000

[7] Chen DaoWen et al., "Systolic Multiple -Valued DTW Processor", China 1991 International Conference on Circuits and Systems, pp. 859-862, June 1991

[8] M. J. Irwin, "A Digit Pipelined Dynamic Time Warp Processor", IEEE Tr. on Acoustics, Speech, and Signal Processing, Vol. 36, No. 9, pp. 1412-1422, Sept. 1988



### 저 자 소 개



안점영(Jeom-Young Ahn)

1964년 2월 한국항공대학교 전자공학과

1979년 2월 동아대학교 전자공학과 (공학석사)

1986년 8월 동아대학교 전자공학과 (공학박사)

1987년 3월 ~ 현재 동의대학교 전자공학과(전공) 교수

※ 관심분야 : 음성인식, 신경망, 디지털 신호처리



최영식(Young-Shig Choi)

1982년 2월 경북대학교 전자공학과 (공학사)

1986년 12월 Texas A&M Univ. 전기공학과(공학석사)

1993년 5월 아리조나 주립대 전기공학과(공학박사)

1987년 2월 ~ 1999년 2월 현대전자 시스템 IC 연구소

1993년 3월 ~ 현재 동의대학교 전자공(전공) 조교수