

한국어 문장-음성 변환 시스템을 위한 하드웨어 및 제어프로그램의 개발에 관한 연구

徐 相 勳

A study on the development of hardware and Control Program for Korean Text to Speech Converter

Seo, Sang-Hoon

목 차

ABSTRACT

I. 서 론

II. 한국어의 음소와 음운 현상

2.1. 한국어의 음소

2.2. 한국어의 음운 현상

III. 한국어의 문장-음성 변환에 대한 분석

3.1. 규칙 합성의 모델

3.2. 한국어 음소의 변화와 운율

IV. 음성 신호 부호화

4.1 PCM 부호화

4.2 u-law companding

V. 문장-음성 변환 시스템의 설계 및 구현

5.1 시스템의 구성

5.2 디지털 신호 처리기 ADSP-2101

5.3 음소 DB의 구성

5.4 하드웨어 시스템의 설계 및 구현

5.5 콘트롤 프로그램의 설계 및 구현

VI. 실험 및 결론

6.1 실험

6.2 결론

참고문헌

부 록

부록 1. 한국어 문장-음성 변환 시스템의 DIP-SW 사양

부록 2. 한국어 문장-음성 변환 시스템의 회로도

Abstract

This thesis presents the TTS(Text To Speech converter) which can convert the Korean text to speech.

The software and hardware, which are main component of TTS, are tentatively designed and made. The two byte combination Korean character code is used for TTS to com-

指導教授 : 朴 長 春

컴퓨터應用專攻 碩士學位 請求論文

municate with host system. And its code should be divided into 3 byte character code, an initial sound, middle sound and final sound for searching the phoenem DB(Data Base).

The divided 3 byte character code is used to create a unique syllable in accord with the stored phoenem DB in EP-ROM.

The stored phoenem DB is consisted of 19 initial consonants, 7 final sounds and 10 vowels, and used by control program to make the process of vocal sound.

The ADSP-2101KG50 manufactured by Analog devices, Inc. is used as a main CPU of TTS. The ADSP-2101KG50 is DSP chip for processing of high speed arithmetic and digital signal and can execute all of instruction using the one clock cycle.

The method of communication between the host and TTS is RS-232C, and for this communication, the UM6551 ACIA from UMC company is used.

TTS can transfer voice message on real time via the low-speed communication path (less 100 bps). But it needs to increase voice quality to take advantage of the improvement in phoenem DB, for processing of vocal sound and function of rhythm.

I. 서론

언어는 인간이 의사를 전달하기 위한 수단으로 그 중요성은 누누이 강조되었으나 음성에 관한 자연 과학적 연구는 전기 통신에 의한 음성 전달이 가능해지면서 시작 되었다. 근래에는 컴퓨터 보급의 확산과 더불어 음성을 이용한 인간과 기기간의 정보 교환에 대한 연구가 활발히 진행되어 최근 30년간에 현저하게 진보를 하였다. 이중 음성합성 분야는 각국에서 자국어에 대한 연구가 활발히 진행되고 있으며 국내에서도 한국 전자통신 연구소를 비롯하여 몇몇 연구기관에서 한국어에 대한 연구가 활발히 진행되고 있다.

문장을 입력하여 그것을 음성으로 변환하는 것을 문장-음성 변환이라하며, 여기에 사용되는 음성 합성 방법을 규칙 합성이라한다. 규칙 합성에는 음원과 조음의 파라미터계열을 직접 자연음성의 데이터를 사용하지 않고 완전히 규칙에 의해 생성하여 합성하는 순수합성 방식과, 조음 파라미터계열의 일부를 직접 자연음성의 데이터를 사용하여 음성을 합성하는 분석합성 방식이 있다.[19]

원리적으로는 순수합성 방식이 이상적인 방식이지만 조음 파라미터를 생성하기 위한 규칙과

음성 기본단위의 데이터를 생성하기가 매우 어렵다. 이에 대하여, 분석합성계를 이용하는 규칙 합성 방식은 적어도 단위음성의 데이터로는 자연음성으로부터 직접 구한 것을 사용할 수 있기 때문에 비교적 간단한 시스템으로도 좋은 음질을 얻을 수 있다.

음성 신호를 디지털 형태로 표현하는 방법에는 아날로그 음성 신호를 샘플링과 양자화 과정을 통하여 부호화된 신호를 얻는 파형 부호화 방식과 합성을 위한 모델의 파라미터들로 음성 신호를 나타내는 스펙트럼 부호화 방식이 있다.

파형 부호화 방식을 이용한 음성 합성에서는 음성 신호를 큰 오차 없이 디지털 형태로 변환한 후 합성에 사용하므로 합성음질이 깨끗하고 또한 디지털화의 과정이 간단하나 데이터율이 15,000~200,000 bits/sec로 매우 크다. 반면 스펙트럼 부호화 방식은 음성신호의 디지털화 과정이 복잡하고 합성 음질이 파형 부호화 방식에 비하여 다소 떨어지지만 데이터율이 75~15,000bits/sec로 매우 적다.[17]

따라서 파형 부호화 방식에 의하여 부호화된 음성 정보를 직렬로 전송, 재 합성 하기 위하여는 수만 bps의 전송 속도가 요구되나, 문장-음성 변환 시스템을 이용하여 합성할 경우는 수백 bps의 전송 속도로도 합성음을 출력하는데 충분하다. 다만 재 합성의 경우는 자연의 모든 소리를 합성할 수 있으며 음질이 우수하나, 문장-음성 변환시스템은 문장으로 표기할 수 있는 음성만을 합성할 수 있고 재 합성에 비해 음질이 떨어진다.

본 논문에서는 직렬 전송로를 통하여 수신된 문장으로부터 파형 부호화 방식에 의하여 저장된 음소를 합성, 한국어로 출력하기 위한 시스템의 하드웨어와 제어 프로그램을 설계하고 시험 제작하였으며, 이로써 저속의 직렬 통신 선로를 통해서도 합성 음성을 출력할 수 있도록 하였다.

주 프로세서로는 DSP 칩인 ANALOG DEVICE사의 ADSP-2101KG50을 이용하였고, 직렬 통신 기능을 위하여 UMC사의 UM6551을 사용하였다. 제어 프로그램은 ADSP-2101 어셈블리어로 작성하여 어셈블한 후 27C128 EP-ROM에 저장(fusing)하였고, 음소 DB는 음소단위 데이터를 8비트의 u-law 컴팬딩으로 추출한 후 27C040 EP-ROM에 저장하여 보드에 장착하였다. NATIONAL사의 TP3054 CODEC은 A/D 변환과 대역 필터 기능을 제공하여 u-law 컴팬딩 코드로 부터 합성음을 생성한다.

Ⅱ. 한국어의 음소와 음운 현상

2.1. 한국어의 음소

한국어의 음소는 자음과 모음이 있으며 이들 음소는 서로 결합하여 음절을 이룬다. 자음 음소는 평음과 격음 및 경음을 합하여 19개가 있고 모음 음소는 10개의 단모음이 있다.

자음에는 유성자음과 무성자음이 있으나 모음은 모두가 유성음이다.

표 2.1은 자음을 조음 위치와 조음 방법에 따라 분류 한 것이다.

〈표 2.1〉 조음 방법과 조음 위치에 따른 한국어의 자음 음소

조음방법 \ 조음위치		양순음	치경음	구개음	연구개음	성문음
파열음 마찰음		ㅂ	ㄷ ㅅ	ㅈ	ㄱ	
		ㅍ	ㅌ	ㅊ	ㅋ	ㅎ
		ㅃ	ㄸ ㅆ	ㅉ	ㄲ	
공명음	비음	ㅁ	ㄴ		ㅇ	
	유음		ㄹ			

이들 자음 음소가운데 말음 규칙에 의하여 종성으로 사용될 수 있는 것은 ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅇ의 7개 음소이며 기타의 글자소는 절음 법칙에 의하여 표 2.2와 같이 이들 7개 가운데 하나로 발음된다.

〈표 2.2〉 절음법칙에 의한 받침의 발음

종성글자소	대표음
ㄱ, ㄲ, ㅋ	ㄱ
ㄴ	ㄴ
ㄷ, ㅌ, ㅆ, ㅈ, ㅊ, ㅉ	ㄷ
ㄹ	ㄹ
ㅁ	ㅁ
ㅂ, ㅍ	ㅂ
ㅇ	ㅇ

모음 음소에는 10개의 단모음과 단모음의 조합으로 이루어지는 11개의 복모음이 있다. 표 2.3은 10개의 단모음을 조음 방법과 조음 위치에 따라 분류 한 것이다.

〈표 2.3〉 조음 방법과 조음 위치에 따른 한국어의 모음 음소

조음방법 \ 조음위치	전설모음	중 설 모 음		후설모음
폐 모 음	ㅣ	거	ㅡ	ㅓ
반 개 모 음	ㅐ	ㅑ	ㅕ	ㅗ
개 모 음	ㅔ		ㅗ	

복모음에는 ㅑ ㅕ ㅑ ㅕ ㅑ ㅕ ㅑ ㅕ ㅑ ㅕ ㅑ ㅕ 등의 11개가 있으며 이들은 선행되는 모음에 따라 다음의 4가지로 분류된다.

- ㅣ 선행모음 ㅑ(ㅣ+ㅑ), ㅕ(ㅣ+ㅕ), ㅑ(ㅣ+ㅑ), ㅕ(ㅣ+ㅕ)
 ㅑ(ㅣ+ㅑ), ㅕ(ㅣ+ㅕ)
 ㅑ 선행모음 ㅑ(ㅑ+ㅑ), ㅕ(ㅑ+ㅕ)
 ㅕ 선행모음 ㅑ(ㅕ+ㅑ), ㅕ(ㅕ+ㅕ)
 ㅡ 선행모음 ㅑ(ㅡ+ㅑ)

2.2. 한국어의 음운 현상

음운 현상은 조음의 편이화 원리 또는 최소 노력 원리등에 의하여 발음을 최대한 경제적으로 하려는 데서, 또는 가장 편이하게 하려는 데서 나타난다.[11]

한국어에서는

절음법칙

연음법칙

구개음화

자음동화

경음화

음운첨가

자음축약 등의 음운 현상이 나타난다.

2.2.1. 절음법칙과 연음법칙

한국어의 모든 자음은 초성으로 쓰일 수 있으나 말음으로는 ‘ㄱ, ㄴ, ㄷ, ㄹ, ㅁ, ㅂ, ㅇ’의 7가지만 쓸 수 있다. 따라서 받침 글자로 사용되는 글자소 가운데 ‘ㄱ, ㅋ’, ‘ㅅ, ㅆ, ㅈ, ㅊ, ㅌ’, ‘ㅇ’ 등은 각각 [ㄱ, ㄷ, ㅂ]의 대표음으로 발음 되는데 이를 절음 법칙이라 한다.

또, 받침으로 끝나는 말에 모음으로 시작되는 말이 연이어 오면 앞말의 끝 자음을 뒷말의 첫 음으로 발음한다.

[보기]	덜다	-----	[덜따]
	웃이	-----	[오시]

다만, 받침으로 끝나는 말에 연이어 오는 말이 ‘ㄱ, ㅋ, ㄴ, ㄷ, ㅂ’들로 시작하는 실질 형태소인 경우는 받침을 대표음으로 바꾸어서 뒤 음절의 첫소리로 발음 한다.

[보기]	웃어른	-----	[우더른]
	맛없다	-----	[마덜다]

2.2.3. 자음동화

음절의 종성이 다음 초성의 영향을 받아서, 어느 한쪽 또는 양쪽이 그와 같거나 비슷한 소리로 바뀌는 현상으로, 양 자음에 비음 또는 유음이 포함되어 있을 때 일어난다. 자음 동화중 예외없이 일어나는 현상을 ‘절대 동화’라 하고 화자의 습관이나 기호에 따라 일어날 수 있는 현상을 ‘수의적 동화’라 하나 수의적 동화는 표준 발음으로 인정하지 않는다.

[절대동화의 보기]		[수의적동화의 보기]	
난로	----- [날로]	감기	----- [감기](○)/[강기](×)
천리	----- [첼리]	젓먹이	----- [전머기](○)/[점머기](×)

2.2.4. 경음화

두개의 무성음이 연속될 때 앞 음절 끝 자음의 영향을 받아서 뒤 음절의 첫소리가 경음으로 발음되는 현상으로, 무성음 ‘ㄱ, ㄷ, ㅂ’뒤에 오는 무성음 ‘ㄱ, ㄷ, ㅂ, ㅅ, ㅈ’은 경음으로 발음한다.

[보기]	웃고름	-----	온고름	-----	[오꼬름]
	있다	-----	인다	-----	[이따]
	곱돌	-----		-----	[곱톨]

한자어중 ‘ㄹ’중성 뒤에 오는 ‘ㄷ, ㅅ, ㅈ’등도 경음으로 발음한다.

2.2.5. 음운첨가

두개의 형태소가 합쳐져 합성어와 파생어를 이룰 때 발음의 편리를 위하여 음을 첨가하여 발음 한다. 합성어와 파생어에서, 앞 단어나 접두사의 끝이 자음이고 뒤의 단어나 접미사의 첫 음절이 ‘ㅣ’선행 모음인 경우에는 ‘ㄴ’음을 첨가하여 발음한다.

[보기]	숨+이불	-----	[숨니불]
	한+여름	-----	[한너름]

2.2.6. 자음축약(격음화)

종성‘ㅎ’의 발음은 그 나타나는 조건에 따라 다음 4가지 가운데 하나로 발음된다.

- i) ‘ㅎ’+‘ㄱ, ㄷ, ㅌ’의 경우는 두 음소를 합하여 ‘ㅋ, ㅌ, ㅍ’으로 발음한다.
- ii) ‘ㅎ’+‘ㅅ’의 경우는 두 음소를 합하여 ‘ㅆ’으로 발음 한다.
- iii) ‘ㅎ’+‘ㄴ’의 경우는 ‘ㅎ’을 ‘ㄴ’으로 발음 한다.
- iv) ‘ㅎ’+‘모음으로 시작하는 어미 또는 접미사’의 경우는 ‘ㅎ’을 발음하지 않는다.

[보기]	i) 각하	-----	[가카]
	ii) 많소	-----	[만쏘]
	iii) 놓는	-----	[논는]
	iv) 많아	-----	[마나]

Ⅲ. 한국어의 문장-음성 변환에 대한 분석

3.1. 규칙 합성의 모델

어떠한 문장이라도 그에 대응하는 합성음을 출력 하도록하는 무제한 음성의 규칙 합성 시스템은 크게 언어처리와 음성합성 기능으로 이루어진다.

언어 처리기능은 입력문을 대상으로 구문 해석을 수행하여 음소의 지속 시간과 핏치(pitch), 휴지(pause), 강세(accent)등에 관한 운율 기호열을 생성하고 각 어절에서의 음운 변동과 숫자, 약어등에 대한 음운 기호열을 생성한다.

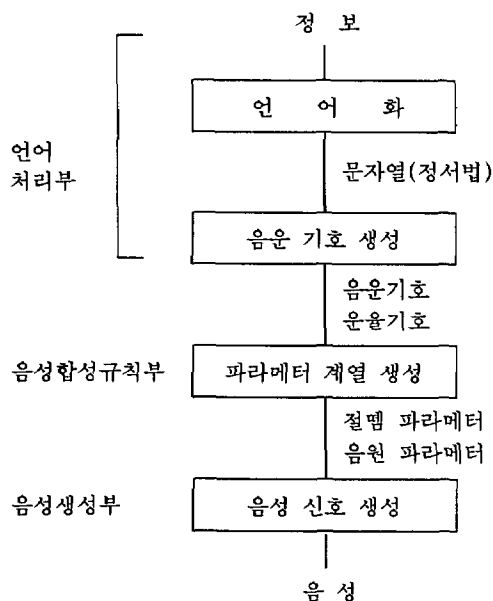
음성 합성 기능은 언어처리 기능에 의해 생성된 운율 기호와 음운 기호열로부터 미리 저장된 단위 음성과 제어규칙에 의해 음성을 합성한다.

문장-음성 변환을 위한 단위 음성으로는 음소, 반음절, 음절, 단어등을 사용할 수 있으며 이들의 특징은 표 3.1과 같다.

〈표 3.1〉 단위 음성의 특징

단 위 음 성	데 이 터 량	특 징
음 소	적 다	인접하는 음 사이의 천이를 실현하기 어렵다.
반 음 절	음소와 음절의 중간 정도	인접하는 음 사이의 천이를 유지할수 있어 접속점에서 스펙트럼의 불연속을 최소화 할 수 있다.
음 절	많 다	타이밍 제어등이 어렵다.
단 어	대단히 많다	고품질의 음성을 생성할 수 있다.

정서법으로 쓰여진 문장으로부터 음성을 생성해 내는 것을 문장-음성 변환이라 하고, 이때 사용되는 음성 합성 방법을 규칙합성이라 한다. 그림 3.1은 정서법으로 쓰여진 임의의 문장에 대해 음성을 생성하기 위한 모델을 나타내고 있다.



[그림 3.1] 규칙합성에 의한 음성생성과정의 모델

3.2. 한국어 음소의 변화와 운율

3.2.1. 음소의 변화

음의 변화에는 동화나 이화와 같이 여러 가지 요인에 의해 음 체계 일부에 개별적으로 나타나는 조건변화와 환경에 구애되지 않고 나타나는 무조건 변화가 있다. 조건변화가 특정한 음성적 환경을 전제로 함에 대해, 무조건 변화는 어떤 음이 모든 위치에서 다 바뀐다.

이중 조건 변화는 문장-음성 변환 시스템에서 처리되어야 하는 것으로 대부분 음운 법칙에 의하여 처리되나 그중 유성음화 등과 같은 몇몇의 변이 현상은 음운 법칙과는 별도로 처리하여야 한다.

가. 유성음화

자음의 음소가운데 /ㄱ/, /ㄴ/, /ㄷ/, /ㅂ/은 무성음으로 /군인/, /돌/, /바람/ 등에서와 같이 어두 초성에서는 각각 [k], [t], [p]로 발음 되나, /아기/, /버들/, /나비/ 등에서 /ㄱ/, /ㄷ/, /ㅂ/은 각각 [g], [d], [b]로 발음되어 유성음 사이에서 이들 무성음이 유성음화 된다.[18]

또한 '집', '곧'등에서의 받침이 각각 [p'], [t']로 발음되어 조건 변이음이 나타난다.

나. 비음화

모음의 전 또는 후에 비음이 존재할 경우 모음은 비음화 된다.

초성 자음가운데 비음은 /ㄴ/, /ㄹ/이 있으며, 종성으로는 /ㄴ/, /ㄹ/, /ㅇ/이 있어 이들 5개의 자음 전후에 오는 모음은 모두 비음으로 나타난다.

다. 음소 /ㅅ/의 변이

음소 /ㅅ/은 그 출현 위치에 따라 대체로 [s, ʃ]등의 두 가지로 나타난다.

어두 위치나 어중의 단모음이나 반모음 [j]에 선행 할 때는 [s]로 나타나고, 반모음 [w]나 /ㄱ/등에 선행 할 때는 [ʃ]로 나타난다.

3.2.2. 운 율

문장-음성 변환 시스템을 구성하기 위한 운율적 요소로는 강세, 음의 높이, 음의 길이 등이 있다. 강세는 어절의 길이와 종성의 유무에 따라 강세 위치가 결정 된다. 규칙에 의한 강세 위치를 결정하기 위한 조건은 다음과 같다.[13]

- 1) 단음절로 구성된 어절은 항상 강세를 받는다.

- 2) 2또는 3음절 어절인 경우는 처음 나오는 받침 있는 음절이 강세를 받으며 모두 받침이 없는 음절 일 때는 마지막 음절이 강세를 받는다.
- 3) 4음절 이상의 경우 1), 2)에 준하여 첫째 또는 둘째 음절이 강세를 받는다.
- 4) 3음절 이상의 경우는 제2강세를 가진다.
- 5) 경음과 격음은 그 앞의 음절을 받침 있는 음절로 가정한다.

음의 높이는 억양과 강세에 의하여 결정 된다. 억양은 주로 마침표, 쉼표, 느낌표, 물음표 등의 문장 부호에 의하여 결정되며 강세를 받는 음절은 다른 음절 보다 음의 높이가 높다.

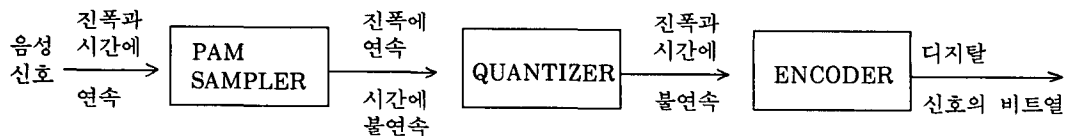
음절의 길이는 한 개의 어절 안에 포함된 음절 수에 반비례하며 모음의 길이가 음절의 길이를 좌우한다.

IV. 음성 신호 부호화

4.1. PCM 부호화

자연인의 음성 신호는 아날로그 신호이나 디지털 회로에서는 디지털 신호 만을 처리할 수 있다. 따라서 디지털 컴퓨터에서 음성 신호를 처리하기 위하여는, 음성 신호를 디지털 값으로 변환하여야 한다.

음성신호의 부호화 방식에는 크게 파형 부호화 방식과 스펙트럼 부호화 방식이 있다. 파형 부호화 방식의 일종인 PCM 데이터를 생성하기 위하여는 PAM 샘플들을 정량화해야 하는데, 그러기 위해서는 각 PAM 펄스의 진폭이 정수값의 근사치로 표시되어야 한다. 음성 신호로부터 PCM 데이터를 생성하기 위한 단계를 도시하면 그림 4.1과 같다.



[그림 4.1] 아날로그 신호의 PCM 코드화 단계

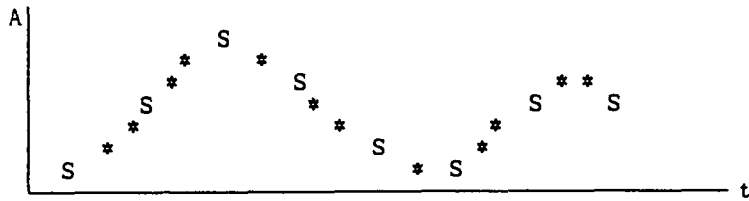
PCM 데이터로부터 음성을 합성하기 위하여는 그림 4.1의 과정을 역으로 수행하게 되나, PAM 펄스를 정량화 함에 따른 정량화 잡음으로 인하여 원래 신호로의 복구는 이루어질 수

없다. 정량화 잡음의 신호대 잡음비는

$$S/N = 6n + 1.8 \text{ [dB]} \dots\dots\dots \text{식 4.1}$$

로 표현되어, 정량화 과정에서 데이터 비트 수를 하나 증가함에 따라 S/N비를 6 dB 증가시킨다.[12]

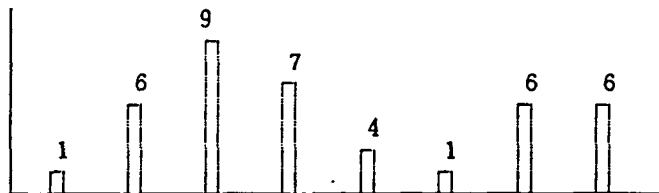
그러나 이와 같은 방법으로 아날로그 신호를 디지털화할 때 각 단계의 값을 균등하게 배분하므로 각 샘플에 대한 평균 절대 오차가 신호의 크기에 관계없이 동일하게 나타나게 된다. 따라서 진폭이 작은 값에 대해서는 상대적으로 많은 오차가 포함 된다.



a) 원래의 아날로그 신호



b) 샘플링된 PAM 신호 (T_s : 샘플링 주기)



c) 양자화된 PCM 신호

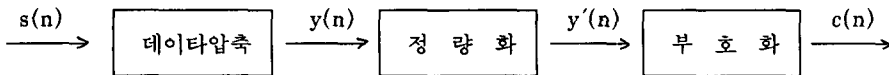
[그림 4.2] 아날로그 신호의 PCM변환

4.2. u-law companding

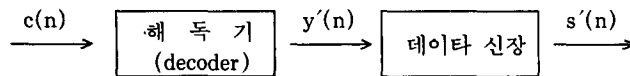
PCM에서 각 단계의 값을 균등하게 배분하여 발생하는 문제는 진폭이 작은 값에 대해서는 많은 정량화 단계를 두고, 큰 진폭에 대해서는 적은 정량화 단계를 두도록하는 비선형 코딩으로 개선이 가능하다.

컴팽딩(companding : compressing-expanding)은 균일한 정량화 단계를 두고 신호를 압축 팽창을 하여 비선형 코딩과 동일한 효과를 얻을 수 있다. 컴팽딩은 log 함수를 이용하여, 입력측의 약한 신호에는 큰 이득을, 강한 신호에는 약한 이득을 줌으로써 신호의 밀도 범위를 압축시키는 과정이다. A-law와 u-law가 가장 일반적인 컴팽딩 방법이며, u-law는 북미의 디지털 통신에 사용되고, A-law는 유럽 지역의 디지털 통신에 사용되고있다.[3]

그림 4.3은 컴팽딩 방식을 이용한 신호 변환 과정을 나타내고 있다.



a) 아날로그 신호의 디지털 신호 변환



b) 디지털 신호의 아날로그 신호 변환

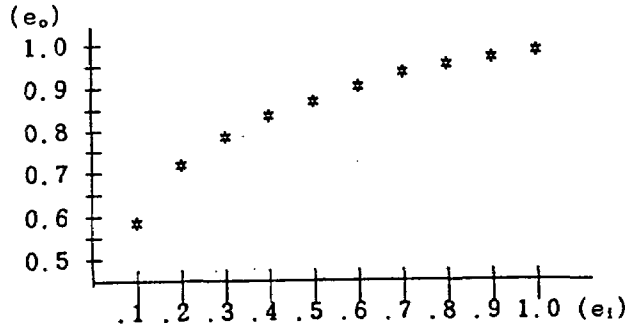
[그림 4.3] 컴팽딩 방식에 의한 신호의 변환 과정

식 4.2는 u-law 양자화를 위한 기본 수식이며 u값은 부호화된 값의 단계별 크기에 영향을 주는 상수로 북미지역의 디지털 통신에서는 255를 사용하고 있다.

$$e_o = \frac{\log(1+ue_i)}{\log(1+u)} \dots\dots\dots \text{식 4.2}$$

$$(0 \leq e_i, e_o \leq 1)$$

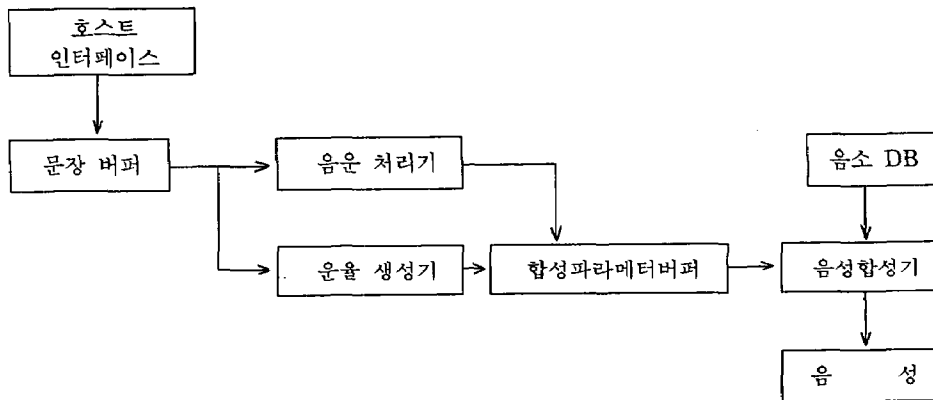
그림 4.4는 u 값을 255로 하였을때 신호 e_i 가 식 4.2에 의하여 u-law 컴팽딩 데이터로 변환된 값을 그래프로 나타낸 것이다.

[그림 4.4] u-law 컴펜딩에 의한 부호화 그래프 ($u=255$)

V. 문장-음성 변환 시스템의 설계 및 구현

5.1. 시스템의 구성

그림 5.1은 본 논문에서 구축하고자 하는 문장-음성 변환 시스템의 구성도이다. 호스트 인터페이스 부분에서 수신한 문자열은, 음운 처리기와 운율 처리기에 의하여 음운 파라미터와 운율 파라미터로 변환되어 합성 파라미터 버퍼에 저장된다. 음성 합성기는 버퍼에 있는 합성 파라미터를 분석하고, 음소 DB에서 해당 음소를 읽어 합성음을 생성한다.



[그림 5.1] 문장-음성 변환 시스템의 구성도

5.2. 디지털 신호 처리기 ADSP-2101

ADSP-2101은 고속의 산술 연산과 디지털 신호를 처리할 수 있는 디지털 신호 처리기(DSP)로, 16비트 데이터 메모리 1KW와 24비트 프로그램 메모리 2KW를 내장하고 있다. 내부의 데이터 버스는 데이터 메모리 데이터 버스와 프로그램 메모리 데이터 버스가 독립되어 있어, 데이터 메모리 관계 명령과 프로그램 메모리 관계 명령을 동시에 처리할 수 있다. 모든 명령어는 단일 클럭으로 처리되며, DO UNTIL 명령은 보다 효율적인 프로그램 루프를 수행할 수 있다.

주된 구성 요소로는

산술 및 논리 연산기(ALU)

곱셈기(MAC)

자리 이동기(BARREL SHIFTER)

2개의 어드레스 생성기(DAG1, DAG2)

타이머

직렬 통신 포트

등이 있다.

5.2.1. 산술 및 논리 연산기 (ALU)

ALU는 두개의 X 레지스터 AX0, AX1과 두개의 Y 레지스터 AY0, AY1 그리고 AR 레지스터와 AF 레지스터를 요소로 하는 2개의 레지스터 뱅크로 구성되어 있다. 이들 레지스터는 MSTAT의 비트 0에 의하여 두개의 뱅크중 하나를 이용하게 된다. ALU는 xop와 yop를 오퍼랜드로하여 산술 연산 또는 논리 연산을 수행하고 그 결과를 R 또는 F 레지스터에 저장한다.

xop로는

AX0, AX1, AR, MR0, MR1, MR2, SRO, SR1 등이 사용 가능하며

yop로는

AY0, AY1, AF가 가능하다.

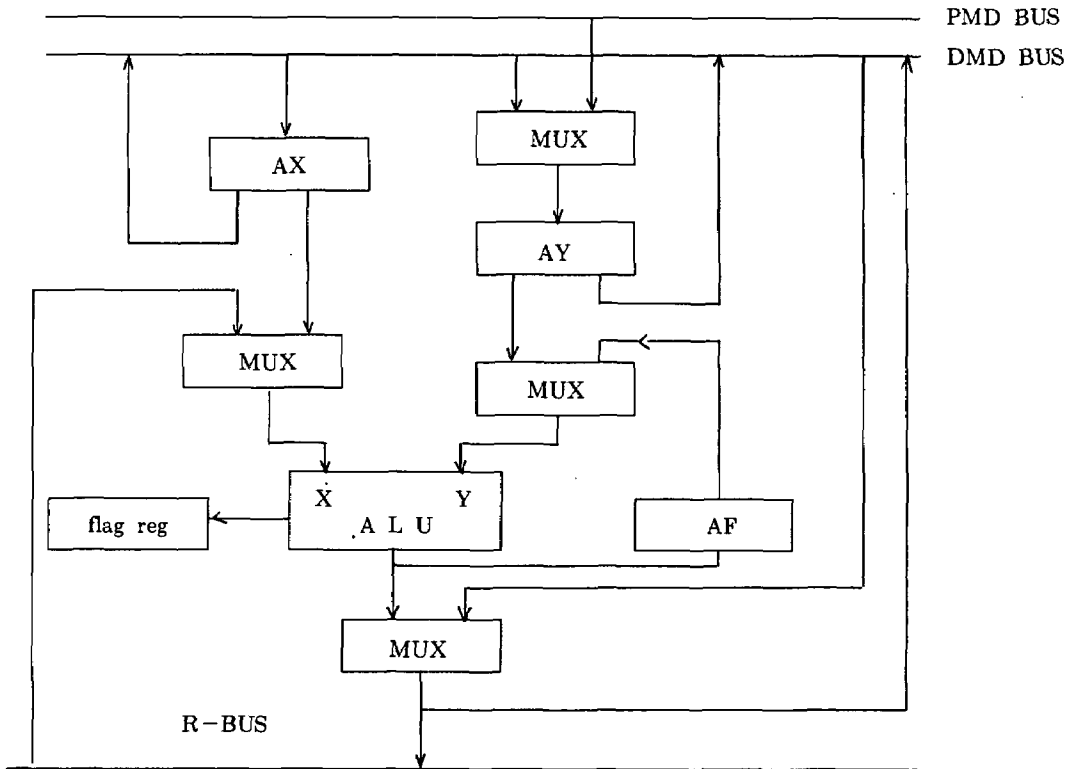
ALU에서 처리 가능한 연산으로는

xop와 yop간의 가감산(carry flag 포함 가능) 및 논리 연산

yop의 increment/decrement

xop 또는 yop의 2의 보수 생성

등이 있다. 그림 5.2는 ALU의 블록도이다.



[그림 5.2] ALU의 블록도

5.2.2. 곱셈기 (MAC)

MAC는 오퍼랜드 레지스터로 16비트의 MX0, MX1, MY0, MY1와, 결과 레지스터로 16비트의 MR0와 MR1 그리고 8비트의 MR2로 구성되어 있다.

연산 결과는 최 하위 워드가 MR0에, 그리고 상위 데이터가 MR1과 MR2에 저장된다. 피드백(feed back) 레지스터 MF는 MR1과 연결되어 연산 결과의 비트 16에서 비트 31까지의 데이터를 저장할 수 있다.

MAC에서 처리 가능한 연산의 형식은

$$xop * yop$$

$$MR + xop * yop$$

$$MR - xop * yop$$

이며, 각각 부호 있는 데이터와 부호 없는 데이터에 대하여 연산 가능하다.

xop로는

MX0, MX1, AR, MR0, MR1, MR2, SR0, SR1 등이 사용 가능하고

yop로는

MY0, MY1, MF가 가능하다.

5.2.3. 자리 이동기

자리 이동기는 입력된 값을 32비트까지 단일 클락으로 자리 이동할 수 있는 기능을 가지고 있다. 자리 이동기는 입력 레지스터로 16 비트의 SI가 있으며, 자리 이동 결과를 보관하기 위한 레지스터로 16 비트의 SRO와 SR1이 있다. 또한 자리 이동 횟수를 지정하기 위한 SE 레지스터가 있어 이의 부호에 따라 좌/우 자리 이동이 결정된다.

자리 이동의 횟수를 즉치 값으로 지정할 경우는 주어진 즉치 값의 부호에 따라 자리 이동 방향이 결정된다. 음수 일 때는 오른쪽 방향 자리 이동, 양수 일 때는 왼쪽 방향 자리 이동이다.

자리 이동기에서 처리 가능한 연산은

논리적 자리 이동 (LSHIFT)

산술적 자리 이동 (AShift)

등이며 입력으로 사용 가능한 레지스터는

SI, AR, MR0, MR1, MR2, SR0, SR1

가 있다.

5.2.4. 어드레스 생성기

ADSP-2101에는 2개의 어드레스 생성기 DAG1과 DAG2가 있다. DAG는 간접 번지 지정 기능을 제공하고, 다음에 액세스할 어드레스를 모디파이 레지스터 값 만큼 자동으로 변경한다. DAG에 연관된 레지스터는 I0~I7, M0~M7, L0~L7의 24개가 있다.

I 레지스터는 액세스할 메모리의 어드레스를 가지고 있으며, 메모리 액세스 후 M 레지스터의 값이 자동으로 더해진다. 환형 큐등과 같이 메모리 버퍼를 환형으로 사용하기 위해서는 L 레지스터에 버퍼의 크기를 설정해주어야 하며 환형 버퍼를 액세스한 후의 I 레지스터 값은

$$(I + M - \text{버퍼의 시작주소}) \text{ MOD } L + \text{버퍼의 시작 주소} \dots\dots\dots \text{식 5.1}$$

로 되어 환형 버퍼의 영역을 벗어나지 않는다.

사용되는 L 레지스터는 I 레지스터와 동일한 첨자 번호를 갖는 레지스터로 가령 I1에 의한

메모리 액세스시는 L1이 사용된다.

memory(In, Mm) 형식에의하여 메모리를 액세스하게 되는데, 이때 n과 m은 0~3의 조합 또는 4~7의 조합으로, 동일한 DAG에 사용되는 레지스터 이어야 한다. I0-I3, M0-M3, L0-L3은 DAG1 군이고, 그 나머지는 DAG2 군이다.

DAG1에 의하여 생성된 어드레스는 데이터 메모리만 액세스(access)할 수 있으며, DAG2에 의하여 생성된 어드레스는 데이터 메모리와 프로그램 메모리 모두를 액세스 할 수 있다.

5.2.5. 타이머

ADSP-2101의 타이머는 2개의 16트 레지스터 TCOUNT와 TPERIOD 및 8비트 레지스터 TSCALE로 구성되어 있다.

타이머가 동작 가능 상태이면 매 (TSCALE + 1)회의 클락 주기 마다 TCOUNT의 값이 1 감소 되며, 0이 되었을 때 TPERIOD 레지스터의 값을 TCOUNT에 재 로드하고, 타이머 인터럽트가 발생 가능한 상태이면 인터럽트를 발생 한다. 타이머 인터럽트의 발생 주기는 식 5.2 와 식 5.3에 의하여 구할 수 있다.

$$\text{타이머 인터럽트 주기} = (\text{TPERIOD} + 1) * (\text{TSCALE} + 1) \text{ cycle} \dots\dots\dots\text{식 5.2}$$

$$\text{첫 인터럽트 발생 주기} = (\text{TCOUNT} + 1) * (\text{TSCALE} + 1) \text{ cycle} \dots\dots\dots\text{식 5.3}$$

5.2.6. 직렬 통신 포트

ADSP-2101은 SPORT0과 SPORT1 2개의 직렬 통신 포트를 가지고 있으며, 각각의 직렬 통신 포트에는 다음 5개의 신호 단자가 있다.

· SCLK	-----	serial clock
· RFS	-----	receive frame synchronization
· TFS	-----	transmit frame synchronization
· DR	-----	직렬 데이터 수신단
· DT	-----	직렬 데이터 송신단

SCLK의 주기는

$$\frac{CLKOUT}{2} * (SCLKDIV + 1) \dots\dots\dots\text{식 5.4}$$

에 의하여 산출되며, 표 5.1은 CPU 클락 주파수가 12.288MHz일 때 SCLKDIV 값과 그에 대한 SCLK를 나타내었다.

SPORT를 통하여 송수신되는 데이터의 길이 및 데이터의 형식은 SPORT 제어 레지스터에 설정하는 값에 의하여 결정된다. 송수신 데이터의 길이는 3비트에서 16비트 사이로 설정이 가능하고, 데이터 형식은 A-law 컴팩딩, u-law 컴팩딩, 우측 정렬 부호 확장등이 설정 가능하다.

〈표 5.1〉 SPORT의 SCLKDIV와 SCLK간의 관계

S C L K D I V	S C L K 주 파 수
20479	300 Hz
5119	1200
639	9600
95	64 KHz
3	1.536 MHz
2	2.048
0	6.144

5.2.7. 메모리 공간

ADSP-2101의 메모리 공간은

데이터 영역(16 비트) : 1KW 내부 RAM

15KW 외부 공간

프로그램 영역(24 비트) : 2KW 내부 RAM

14KW 외부 공간

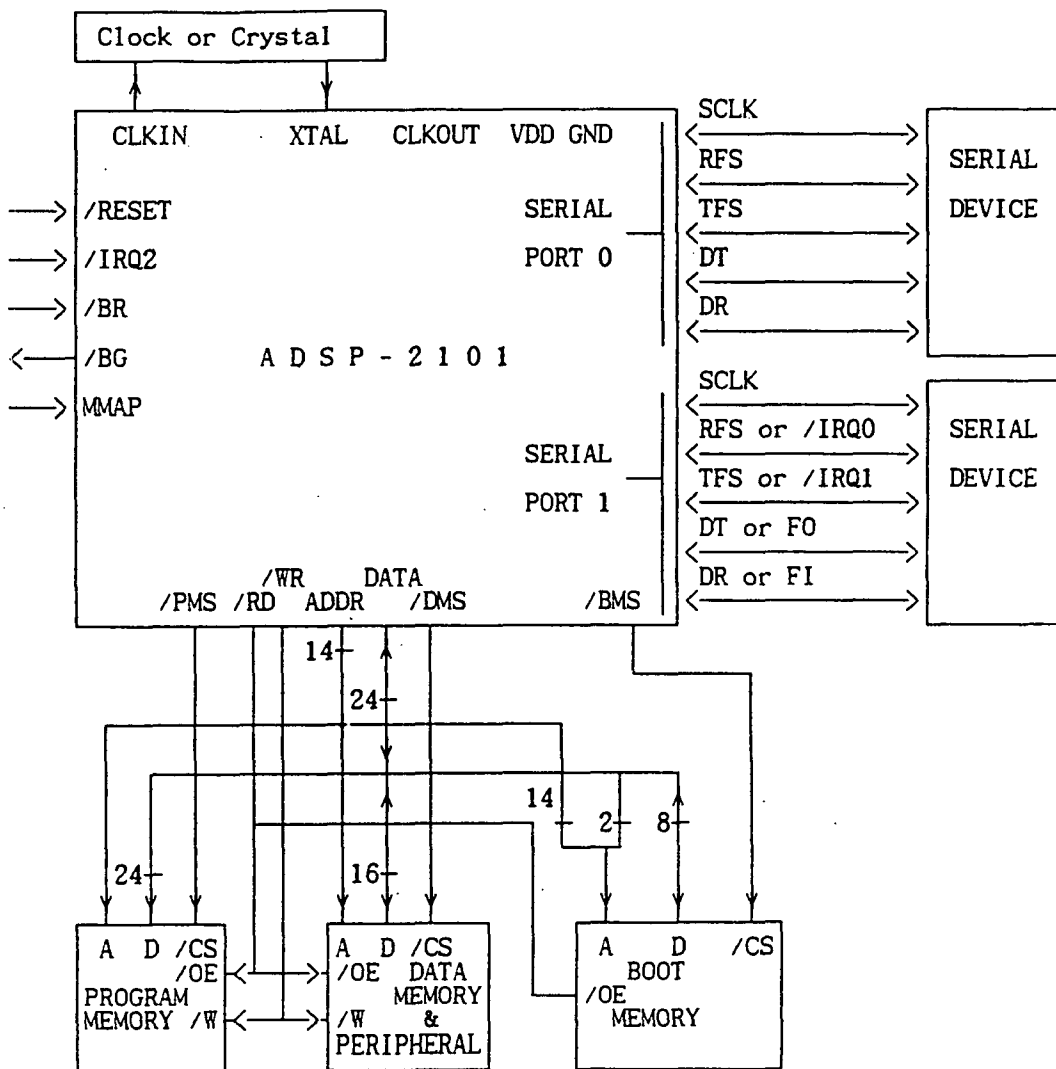
부트 메모리(8비트) : 16KB 외부 공간

를 사용할 수 있다. 부트 메모리는 64KB 까지 사용이 가능하나 하나의 부트 프로그램 모듈의 크기는 16KB로 제한된다.

5.2.8. ADSP-2101을 이용한 기본적인 시스템 인터페이스

ADSP-2101이 외부의 장치들과 인터페이스하기 위한 신호로는 어드레스 버스와 데이터 버스 및 제어 버스가 있다. 외부 장치로는 데이터 메모리 및 주변기기, 프로그램 메모리, 부트 메모리, 직렬 통신 장치등을 들수 있다.

ADSP-2101은 각각 메모리에 사용되는 데이터 버스의 비트를 다르게 하여, 데이터 메모리는 D8-D23의 16비트 단위로, 프로그램 메모리는 D0-D23의 24비트 단위로, 부트 메모리는 D8-D15의 8비트 단위로 액세스 한다. 메모리 공간을 결정하는 어드레스 버스는 14비트로 구성되어 있어, 프로그램 메모리와 데이터 메모리를 각각 16KW까지 액세스할 수 있다. 부트 메모리는 데이터 버스의 상위 2비트를 A14와 A15로 사용하여 64KB까지 액세스할 수 있다. 그림 5.3은 ADSP-2101을 위한 기본적인 시스템의 블록도를 나타내었다.



[그림 5.3] ADSP-2101을 위한 기본적인 시스템 구성도

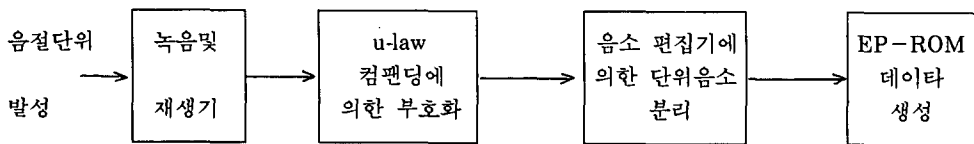
5.3. 음소 DB의 구성

본 논문 구현에서 사용한 합성음의 기본 단위는 음소이다. 음소는 초성, 중성, 종성으로 구분하였으며, 이를 다시 발음 상태등을 고려하여 초성 12별, 중성 19별, 종성 1별로 구성하였다.

19개의 초성 자음은 단모음과 [j], [w]를 포함하여 12개의 모음에 대한 각각의 음소를 가지도록 하여 DB내의 초성 음소는 모두 228개이다.

중성 음소는 단모음 10개와 [j], [w]등 12개를 19초성 자음에 따른 각각의 음소를 가지도록 하여 228개의 중성 음소를 두었으며, 종성은 말음으로 올 수 있는 7개의 음소를 두었다.

음소의 채집은 음절 단위로 발음한 음성을 녹음기에 녹음한 후, 채집할 음소가 포함된 음절을 재생하고 7.5kHz의 샘플링 주기로 u-law 컴펜딩 변환하여 부호화된 데이터를 PC로 전송하였다. PC에서 음소 편집기를 이용하여, 수신된 음절 데이터를 음소 단위로 분리하고 EP-ROM에 저장할 수 있는 데이터 형태로 변환 하였다. 그림 5.4는 본 논문에서 음소 DB를 구성하기 위하여 사용한 처리 과정이다.



[그림 5.4] 음소 DB 구성 단계

음소 DB는 초성과 종성을 하나의 EP-ROM으로 구성하고 중성을 다른 하나의 EP-ROM으로 구성하여 2개의 EP-ROM 27C040에 저장한 후 보드에 장착 하였다. 음소 데이터의 크기는, 하나의 초성 음소가 1KB를 차지하며, 중성과 종성 음소는 각각 2KB의 기억 용량을 차지하고 있다. 음소는 글자 코드로부터 음소 DB내의 데이터 저장 위치를 쉽게 찾아 낼 수 있도록 하기 위하여 프로그램 내부에서 사용하는 글자 코드 순으로 배열하였다. 그림 5.5는 EP-ROM 27C040에 저장된 음소의 배열을 나타내고 있다.

ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅇ

a) 음소 DB 내의 종성 음소의 배열

해 당 자 음	모 음 음 소 의 배 열											
ㄱ	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	[j]	[w]
ㄴ	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	[j]	[w]
ㄷ	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	[j]	[w]
ㄹ	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	[j]	[w]
ㅁ	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	[j]	[w]
ㅂ	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	[j]	[w]
ㅅ	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	[j]	[w]
ㅇ	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	[j]	[w]
ㅈ	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	[j]	[w]
ㅊ	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	[j]	[w]
ㅋ	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	[j]	[w]
ㆁ	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	[j]	[w]

b) 음소 DB 내의 모음 음소의 배열

해 당 모 음	자 음 음 소 의 배 열 순 서															
ㅏ	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㆁ	ㅏ	ㅑ	ㅓ	ㅕ
ㅑ	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㆁ	ㅑ	ㅓ	ㅕ	ㅗ
ㅓ	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㆁ	ㅓ	ㅕ	ㅗ	ㅛ
ㅕ	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㆁ	ㅕ	ㅗ	ㅛ	ㅜ
ㅗ	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㆁ	ㅗ	ㅛ	ㅜ	ㅠ
ㅛ	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㆁ	ㅛ	ㅜ	ㅠ	ㅡ
ㅜ	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㆁ	ㅜ	ㅠ	ㅡ	ㅣ
ㅠ	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㆁ	ㅠ	ㅡ	ㅣ	
ㅡ	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㆁ	ㅡ	ㅣ		
ㅣ	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㆁ	ㅣ			
[j]	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㆁ	[j]			
[w]	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㆁ	[w]			

c) 음소 DB 내의 자음 음소의 배열

[그림 5.5] EP-ROM 27C040에 저장된 음소의 배열

5.4. 하드웨어 시스템의 설계 및 구현

5.4.1. ACIA UM6551의 사양

ACIA UM6551은 RS-232C 방식의 통신 기능을 제공한다. ADSP-2101은 UM6551을 통하여 호스트 데이터를 수신하거나 데이터를 호스트로 전송한다. UM6551의 내부에는 5개의 레지스터가 있으며, 이 레지스터들에 설정되는 값에 따라 전송 속도 및 전송 사양이 선택된다. 표 5.2는 UM6551 내부의 레지스터 맵이다.

〈표 5.2〉 UM6551의 내부 레지스터 맵

RS1	RS0	쓰 기 상 태	읽 기 상 태
0	0	TX 레지스터	RX 레지스터
0	1	소프트웨어 리셋	상태 레지스터
1	0	명 령 레 지 스 터	
1	1	제 어 레 지 스 터	

제어 레지스터에 특정 값을 저장함으로써 워드 길이, 정지 비트 수, 통신 속도등이 설정된다. 저장한 값에 따라 워드 길이는 5, 6, 7, 8비트 중 하나가 선택되고 정지 비트 수는 1또는 2가, 통신 속도는 75 BPS에서 19200 BPS 사이의 15개 전송 속도중 하나가 선택된다.

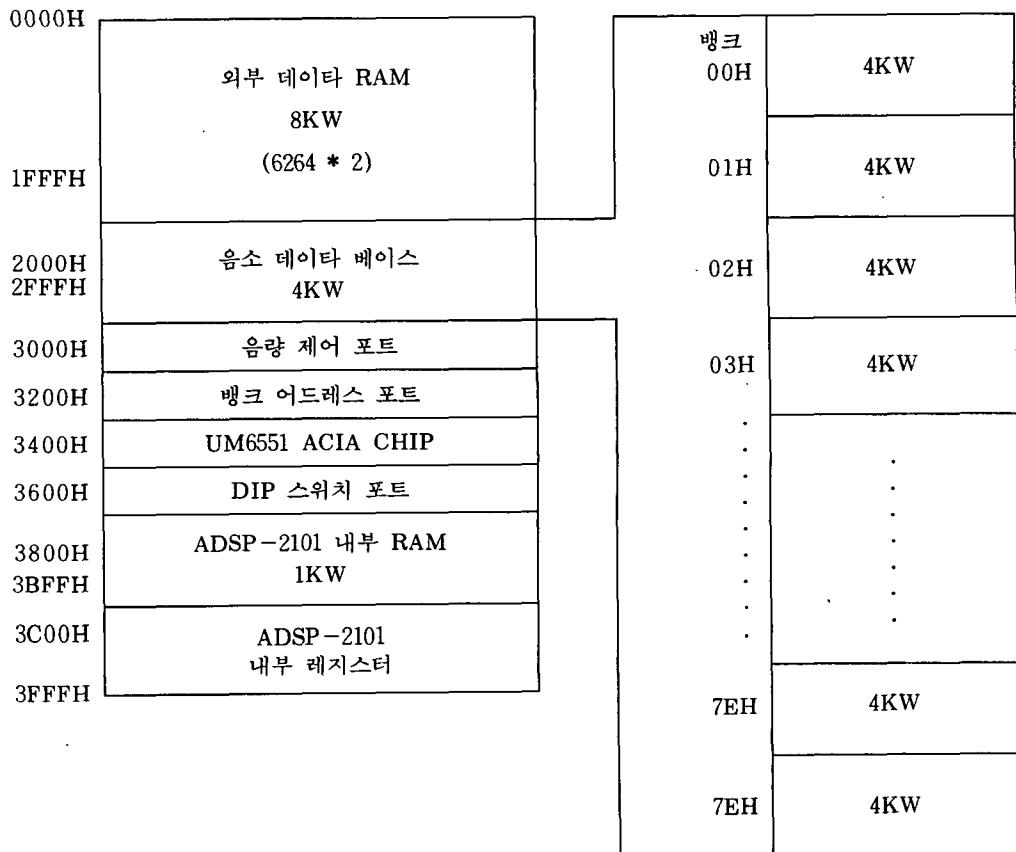
명령 레지스터는 데이터의 정확한 송수신을 위하여 사용되며 이 레지스터에 의하여 패리티 검사 모드가 설정되고, 장치의 준비 또는 비지상태를 알리는 DTR 단자가 제어된다. 인터럽트 신호의 발생 여부도 이 레지스터에 의하여 제어된다.

상태 레지스터는 UM6551의 상태 정보를 가지고 있으며 읽기 전용이다. 본 문장-음성 변환 시스템에서는, ADSP-2101의 IRQ2 단자가 활동 준위일때 이 레지스터를 읽어 UM6551이 인터럽트를 발생한 원인을 분석한다.

TX 레지스터와 RX 레지스터는 각각 쓰기와 읽기 전용이며, 송수신 데이터의 임시 저장 장소로 사용된다. 프로세서가 전송할 데이터를 UM6551로 보내면 TX 레지스터에 저장한 후 비트 0부터 순차적으로 전송한다. RX 레지스터는 수신되는 데이터를 비트 0부터 순차적으로 적재하고, 단위 워드가 모두 수신되면 인터럽트를 발생한다.

5.4.2. 메모리 공간의 할당

프로세서가 여러개의 하드웨어 장치를 사용하여야 하는 경우는 장치간의 데이터 충돌을 피하기 위하여 각각의 메모리 공간을 분리하여야 한다. 본 논문의 문장-음성 변환 시스템에서 사용하고있는 ADSP-2101의 메모리 공간은 데이터, 프로그램 그리고 부트롬을 위한 3개의 독립된 공간을 가지고 있는데 이중 프로그램과 부트 공간은 이들 공간의 모두를 프로그램 롬과 부트 롬에 각각 할당하였고, 데이터 메모리 공간은 외부의 여러 장치들을 사용하기 위하여 분리 하였다. 그림 5.6은 본 논문에서 제작한 문장-음성 변환 시스템의 데이터 메모리 공간 할당도이다. 음소 DB는 ADSP-2101의 메모리 맵 상으로 2000H-2FFFH의 4KW 공간이 할당되어 있으며 워드 데이터중 하위 바이트는 자음 음소 데이터를, 상위 바이트는 모음 음소 데이터를 각각 역세스할 수 있도록 설계 하였다.



[그림 5.6] 문장-음성 변환 시스템의 데이터 메모리 할당

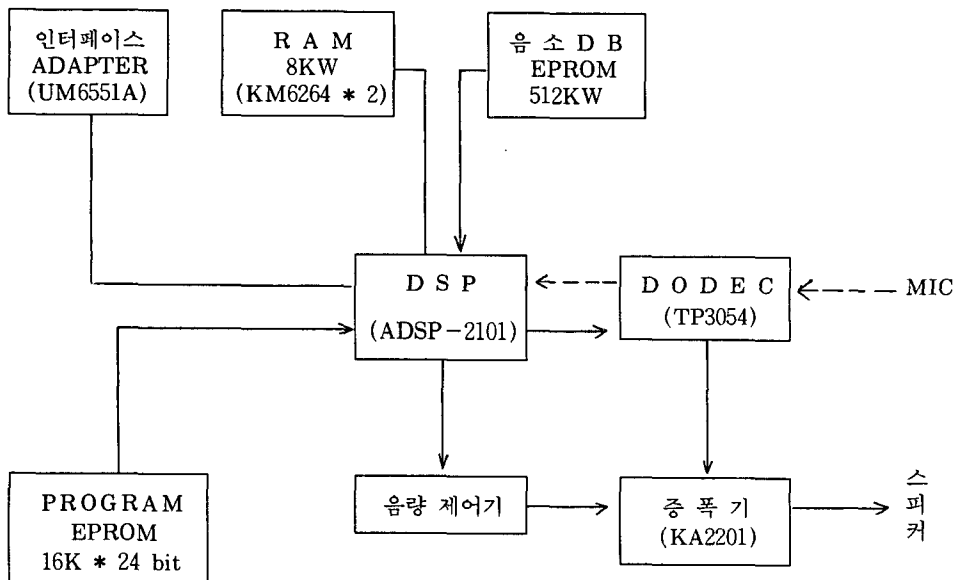
실제 음소 DB를 구성하고 있는 2개의 EP-ROM 27C040의 메모리 공간은 512KW이므로, 4KW만이 할당되어있는 ADSP-2101의 메모리 공간을 통하여 음소 데이터를 액세스하기 위하여는 먼저 뱅크 어드레스를 설정해야 한다. 뱅크 어드레스는 74LS373에 래치(latch)되어 음소 DB의 뱅크 어드레스로 사용된다. 뱅크 어드레스 공간은 00H-7FH의 128개를 두었고 하나의 뱅크 내에는 초성 음소 4개, 중성과 종성 음소 2개씩을 저장하였다.

5.4.2. 하드웨어 시스템의 구성

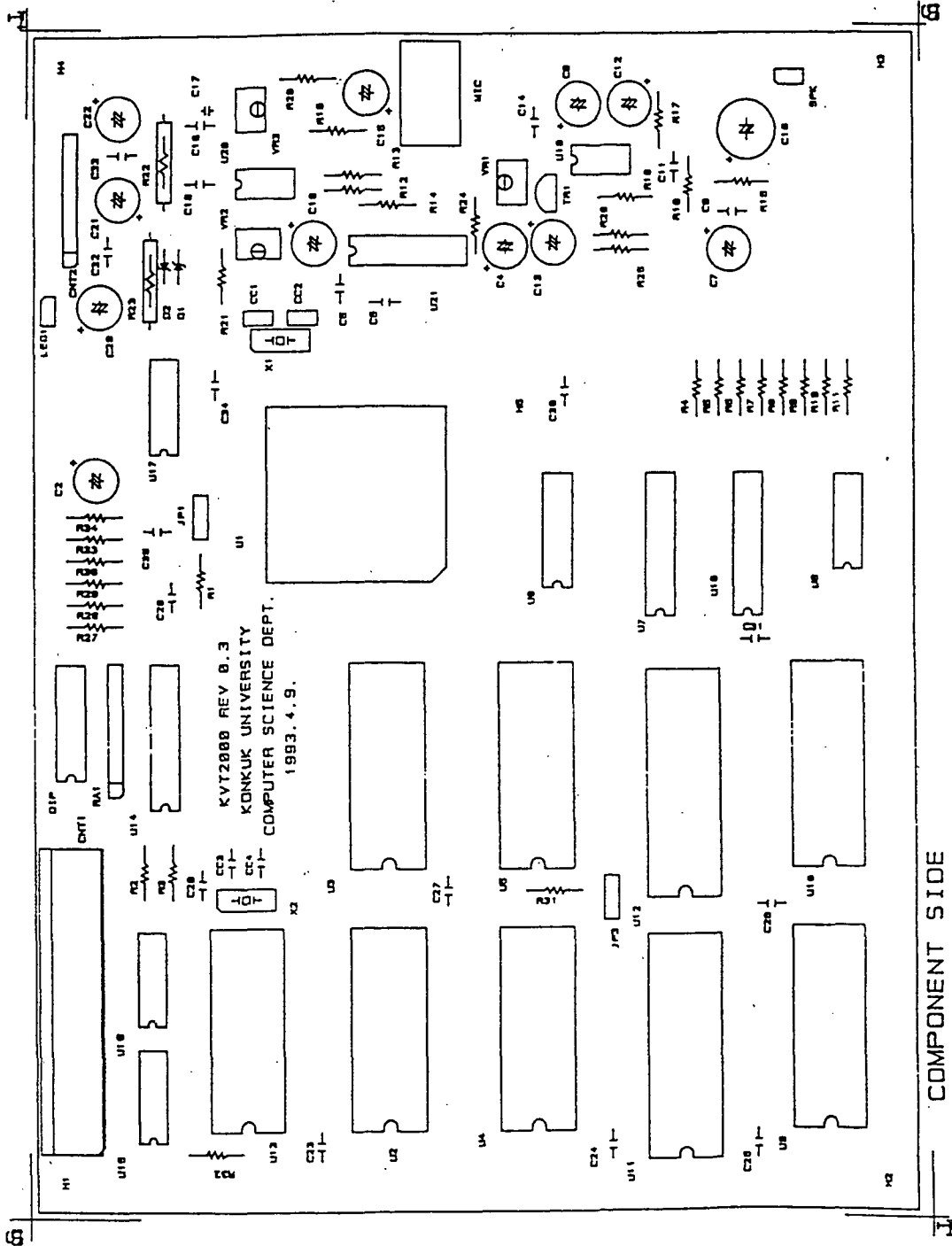
하드웨어 시스템은 ADSP-2101 DSP를 중심으로 하여 호스트 인터페이스 부분, 음소 DB, 음성 합성을 위한 CODEC 부분, 저주파 증폭 부분 등으로 구성되었다.

CODEC TP3054는 DSP로부터 u-law 컴핑딩 코드를 받아 D-A 변환한 후 음성으로 출력한다. 또한 이는 저주파 대역 필터를 내장하고 있어 합성된 음성신호에 포함되어 있는 고주파 성분을 차단시킨다. TP3054에서 출력된 음성 신호로는 직접 스피커를 구동할 수 없으므로, KA2201 저주파 증폭기를 이용하여 증폭 하도록 하였다.

하드웨어의 구현은 PCB를 제작하여 부품을 장착 하였다. 그림 5.7은 본논문에서 구현한 문장-음성 변환 시스템의 하드웨어의 구성도이며 PCB에 실장된 부품의 배치도를 그림 5.8에 나타내었다.



[그림 5.7] 문장-음성 변환 시스템의 하드웨어 구성도

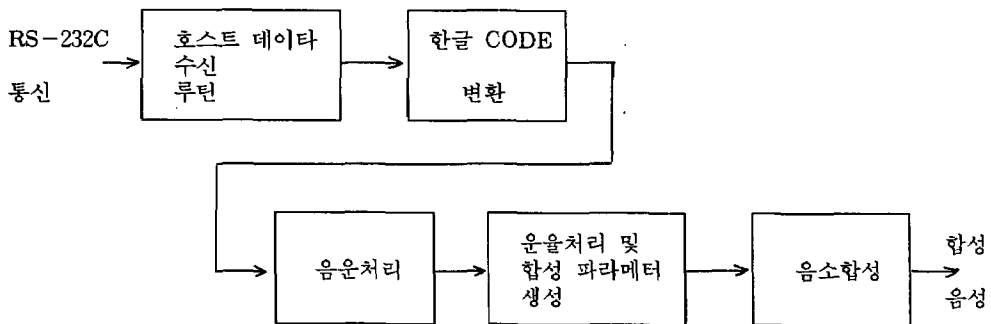


[그림 5.8] 실장된 하드웨어 시스템의 부품 배치도

5.5. 콘트롤 프로그램의 설계및 구현

콘트롤 프로그램은 주 처리부분과 인터럽트 처리 부분으로 구성하였으며 주 처리부분에서는 수신된 문장을 음운및 운율 처리에 의하여 합성 파라미터를 생성하고, 인터럽트 처리 부분에서는 호스트와의 인터페이스및 합성 파라미터에 따른 음성을 합성 하도록 하였다. 그림 5.9는 RS-232C 통신 방식으로 수신된 문자열로부터 합성 음성을 생성하기까지의 흐름을 나타내고 있다.

제어 프로그램은 ADSP-2101의 어셈블리어로 작성하여 어셈블한 후 ROM 데이터 형식으로 변환하고 EP-ROM에 저장하여 보드의 부트 롬 부분에 장착하였다.



[그림 5.9] 문장-음성 변환 프로그램의 흐름도

5.5.1. 주 처리부

주 처리부에는 한글 코드 변환부와 음운 처리부, 운율 처리및 합성 파라미터 생성부가 포함되며 한글 코드 변환 부분에서는 상용 2바이트 조합형 한글 코드로 부터 초성과 중성, 종성을 분류하여 3바이트 형태의 한글 코드로 변환한다. 표 5.3은 호스트와의 인터페이스를 위하여 사용하고 있는 2바이트 조합형 한글 코드표이다.

음운처리는 자음이 모음의 영향을 받아 변환되는 현상과 자음 상호간의 영향에 의하여 변환되는 현상으로 구분하였으며 모음의 영향을 받는 음운현상은 프로그램에서 조건을 검사하여 처리하도록 하였고 이외의 음운현상은 중성과 다음 초성에 대한 변환 테이블(table)을 구성하여 처리 하였다. 음운 처리를 위한 프로그램의 흐름은 다음과 같다.

〈표 3〉 상용 2 바이트 조합형 한글 코드표

C O D E	초 성	중 성	종 성
0 0 0 0 0	*	*	*
0 0 0 0 1	*	*	*
0 0 0 1 0	ㄱ	*	ㄱ
0 0 0 1 1	ㄲ	ㅏ	ㄲ
0 0 1 0 0	ㄴ	ㅑ	ㄴ
0 0 1 0 1	ㄷ	ㅓ	ㄷ
0 0 1 1 0	ㄹ	ㅕ	ㄹ
0 0 1 1 1	ㄺ	ㅗ	ㄺ
0 1 0 0 0	ㅛ	*	ㅛ
0 1 0 0 1	ㅜ	*	ㅜ
0 1 0 1 0	ㅠ	ㅛ	ㅠ
0 1 0 1 1	ㅡ	ㅜ	ㅡ
0 1 1 0 0	ㅟ	ㅛ	ㅟ
0 1 1 0 1	ㅠ	ㅛ	ㅠ
0 1 1 1 0	ㅡ	ㅛ	ㅡ
0 1 1 1 1	ㅢ	ㅛ	ㅢ
1 0 0 0 0	ㅣ	*	ㅣ
1 0 0 0 1	ㅤ	*	ㅤ
1 0 0 1 0	ㅥ	ㅛ	*
1 0 0 1 1	ㅦ	ㅛ	ㅦ
1 0 1 0 0	ㅧ	ㅛ	ㅧ
1 0 1 0 1		ㅛ	ㅨ
1 0 1 1 0		ㅛ	ㅩ
1 0 1 1 1		ㅛ	ㅪ
1 1 0 0 0		*	ㅫ
1 1 0 1 0		*	ㅬ
1 1 0 1 0		ㅭ	ㅭ
1 1 0 1 1		ㅸ	ㅸ
1 1 1 0 0		ㅹ	ㅹ
1 1 1 0 1		ㅺ	ㅺ
1 1 1 1 0			

· start

```

if(현재의 음절이 유종성 음절이고 다음 음절의 초성이 'ㅇ'){ .....①
    if(다음 음절의 중성이 'ㅣ'){ .....②
        if(현재 음절의 종성 = 'ㄷ')
            종성을 'ㅈ'으로 변환
        else if(현재 음절의 종성이 'ㅌ')
            종성을 'ㅊ'으로 변환
    } else if(다음 음절의 중성이 (ㅌ, ㄴ, ㄷ, ㅌ, ㄴ)중 하나).....③
        말음법칙에 의한 종성의 귀착
    }

```

변환 table에 의한 종성과 초성의 음운처리

· end

프로그램 문장 ①은 종성이 모음의 영향을 받아 변화되는 경우의 다음 초성은 'ㅇ' 이므로 이를 검사하기 위한 것이다. ②는 구개음화 조건을 검사하고, ③은 종성을 귀착 처리한 후 다음 음절의 초성으로 연음 처리해야 하는 조건을 검사한다.

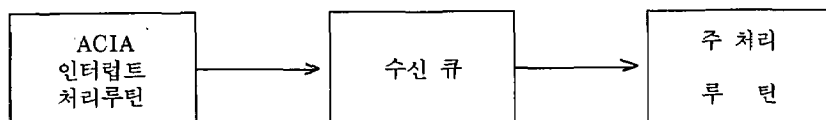
운율 처리 루틴은 입력되는 문장 부호와 단어의 강세 규칙에 따라 음량 파라미터, 합성주기 파라미터, 음소 파라미터등을 생성하고 이를 합성 인터럽트 루틴과의 인터이스를 위하여 큐 버퍼에 저장한다. 음의 높낮이는 7.5KHz의 합성 주기를 평음으로 하여, 8KHz를 높은음으로 하고 7KHz를 낮은음으로 하였다. <표 5.4>는 문장 부호에 따른 합성 주기와 합성 음성의 음량을 제어하기위하여 본 논문에서 사용한 값을 나타내었다.

<표 5.4> 문장부호에 따른 합성 주기및 음량 감쇄

문장 부호	합성 주기	음량 감쇄
마 침 표	7.5 KHz	48 / 255
물 음 표	8 KHz	15 / 255
느 깎 표	7 KHz	48 / 255
쉽 표	8 KHz	32 / 255
공 백	7.5 KHz	32 / 255

5.5.2. 인터럽트 처리부

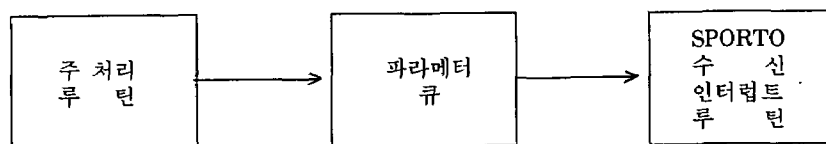
인터럽트 처리부에는 호스트와의 인터페이스 루틴과 합성 파라미터 정보를 읽어 음성을 합성하는 루틴이 있다.



[그림 5.10] ACIA 인터럽트 루틴과 주 처리 루틴 간의 데이터 흐름

호스트 인터페이스 루틴은 UM6551 ACIA의 인터럽트 요구에 의하여 처리가 개시되며, 이때 UM6551의 수신 레지스터에 있는 호스트 데이터는 호스트와 인터페이스를 위하여 마련된 수신 큐에 저장된다. 이 수신 큐의 데이터는 주 처리 루틴에서 순서대로 읽고 처리한다.

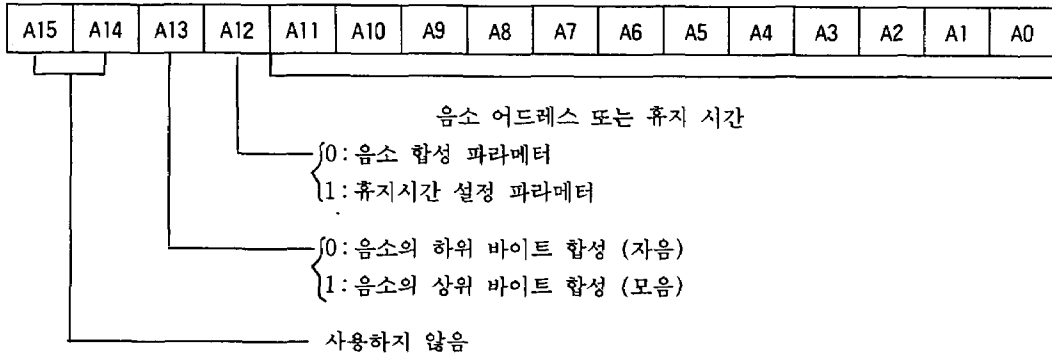
호스트와 원활한 통신을 하기 위하여 수신 큐의 가용 영역이 12 바이트 이하가 되면 X-OFF를 전송하고 100바이트 이상이면 X-ON을 전송하도록 하였다. X-ON 및 X-OFF는 ASCII 코드의 DC1과 DC3을 사용하였다.



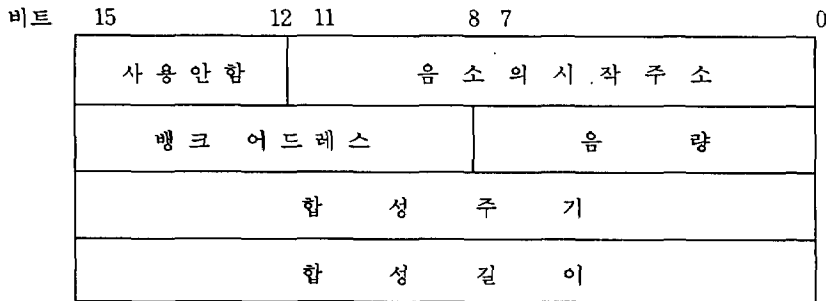
[그림 5.11] 주처리 루틴과 SPORT0 인터럽트 루틴간의 데이터 흐름

음성합성을 위한 인터럽트는 SPORT0 인터럽트를 사용하도록 하였다. SPORT0 수신 인터럽트는 ADSP-2101의 내부 인터럽트로 SPORT0의 수신주기마다 발생하는데, 내부 레지스터의 값을 변경함에 따라 인터럽트 발생주기를 가변할 수 있다.

합성 인터럽트 처리 루틴에서는 파라미터 큐의 데이터를 읽어 파라미터의 성격에 따른 처리를 한다. 합성 파라미터는 유효 음소 합성을 위한 것과 음절간의 휴지시간을 설정하기 위한 2가지의 형태가 있는데 첫번째 파라미터의 비트 12를 검사하여 이를 판별한다.



[그림 5.12] 합성 파라미터의 데이터 형식



[그림 5.13] 합성 파라미터의 구성

파라미터의 비트 12가 1인 경우는 비트 0에서 비트 11까지의 값을 휴지시간 값으로 설정하고, 비트 12가 0인 경우는 파라미터 큐의 다음 3개 워드를 포함하여 음소 합성 정보를 생성한다. 음소 합성 정보를 나타내는 파라미터 버퍼의 4개 워드가 내포하는 의미를 그림 5.13에 나타내었으며 파라미터 버퍼의 데이터로 부터 음성을 합성하는 프로그램의 흐름은 다음과 같다.

· start

```

if(음소 합성 counter < 음소의 합성길이){.....①
    (음소의 시작 주소 + 음소 합성 counter)번지의 값을 읽어 CODEC으로 전송한다.
    음소합성 counter ← 음소 합성 counter + 1
} else if(파라미터 큐 counter ≠ 0){ .....②
    음소 합성 counter ← 0
    addr ← 파라미터 큐(front)
    front ← front + 1
    if(addr의 비트 12 = 0){ .....③

```

```

음소의 시작 주소 ← addr
음소의 뱅크 ← 파라미터 큐(front)
front ← front + 1
합성음의 크기 ← 파라미터 큐(front)
front ← front + 1
합성주기 ← 파라미터 큐(front)
front ← front + 1
음소의 합성길이 ← 파라미터 큐(front)
front ← front + 1
} else { .....④
    음소의 합성길이 ← addr - 0x1000
    음소의 시작 주소 및 뱅크등을 dummy 음소의 위치 값으로 설정
}
파라미터 큐 counter ← 파라미터 큐 counter - 1
· end

```

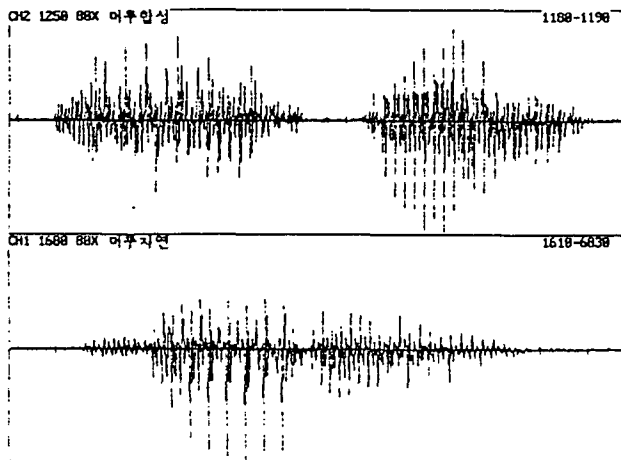
현재 합성 중인 음소에 대하여 합성할 데이터가 남았는지를 ①에서 검사하고, 합성할 데이터가 남았으면 음소 DB로부터 데이터를 읽어 TP3054 CODEC으로 전송 함으로서 합성 처리를 행한다.

문장 ②에서는 counter의 값을 가지고 파라미터 버퍼에 처리해야할 데이터가 있는지를 검사하고 ③에서는 다음에 처리할 버퍼의 내용이 음소 합성 정보인지 휴지시간 설정 정보인지를 판별하여, 음소 합성 정보이면 합성에 필요한 변수들의 값을 설정한다. ④는 휴지시간 설정 부분이다.

VI. 실험 및 결론

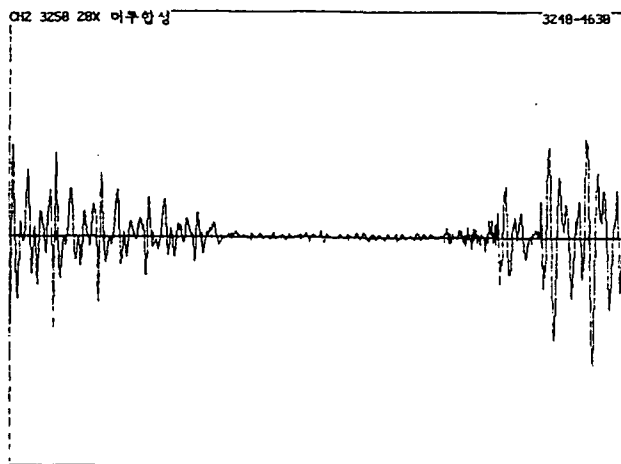
6.1. 실험

본 논문에서 구현한 문장-음성 변환 시스템을 실험 하기 위하여 RS-232C 직력 통신 선로로부터 수신한 문장을 본 시스템으로 합성하고 이를 다시 부호화하여 자연 음의 파형과 비교하였다. 그림 6.1은 '머루'를 합성하였을 때의 파형을 자연음의 파형과 비교한 것이다.

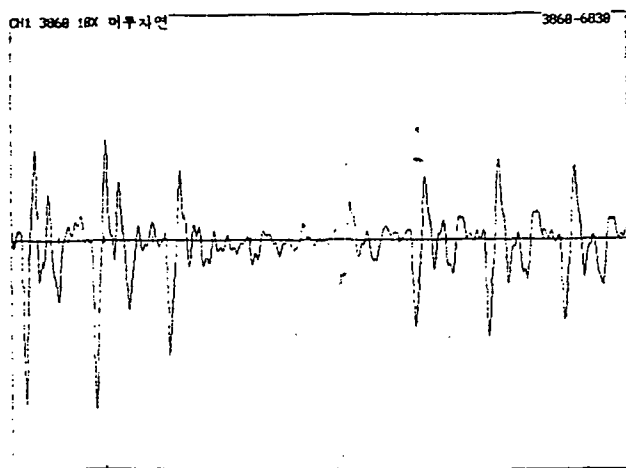


[그림 6.1] '머루'의 합성음(위)과 자연음(아래)의 파형

합성음에서는 음절간의 휴지 기간이 길게 나타나는데 비해 자연음에서는 음절간의 휴지 기간이 매우 짧게 나타나고 있다. 그림 6.2와 그림 6.3은 합성음과 자연음의 음절간의 천이 구간을 확대하여 나타낸 것이다.

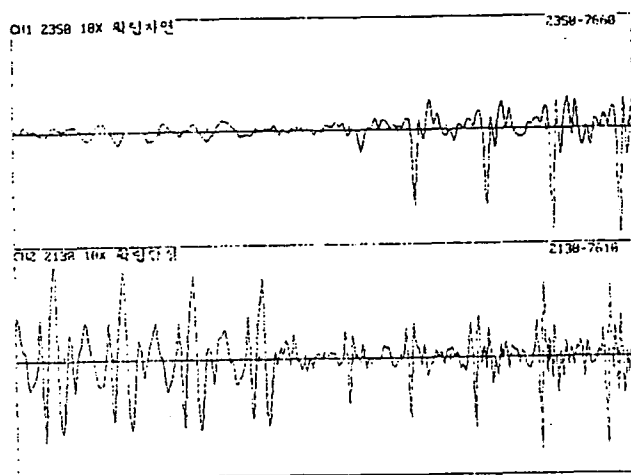


[그림 6.2] '머루'의 음절간 천이(합성음)



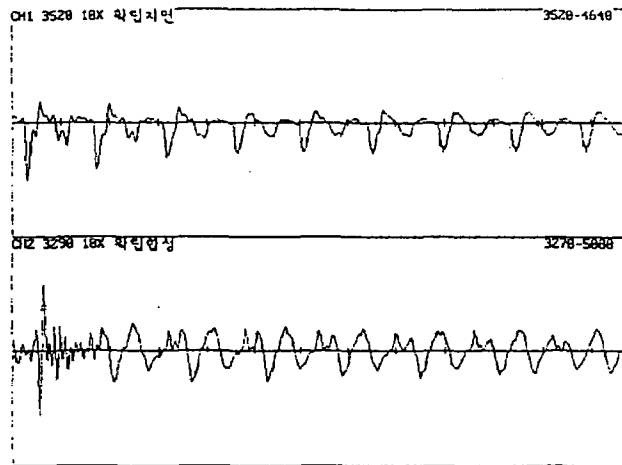
[그림 6.3] '머루' 발성시의 음절간 천이(자연음)

본 논문에서 복모음의 합성은 단모음을 조합하여 처리하도록 하였다. 그림 6.4에 단어 '확립'을 합성할 때 복모음의 합성 구간을 자연음과 비교하여 나타내었다. 자연음에서는 모음 '고'가 약하게 나타나는데 비하여 합성음에서는 강하게 나타나고 있다. 또한 자연 음에서는 '고'에서 '나'로 천이되는 구간에서 진폭이 더욱 약하게 되어 과도구간을 가지고 있으나 합성음에서는 과도구간 없이 모음이 천이된다.



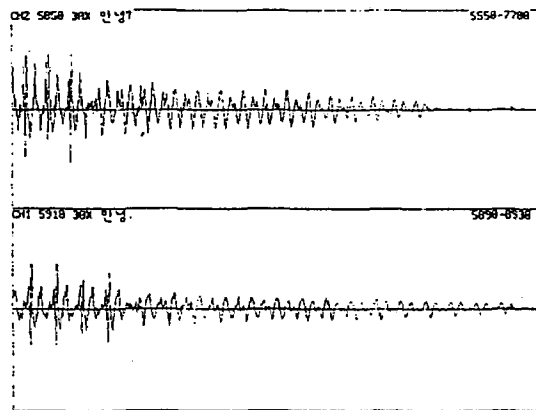
[그림 6.4] 복모음 발성시의 자연음(위)과 합성음(아래)의 파형

단어 '확립'은 자음 동화로 인하여 [황립]으로 발음되며 이는 음운 처리 루틴에서 변환하고 있다. 그림 6.5는 [황립]으로 발음 될때의 'ㅇ'이 발음되는 구간을 나타내고 있다.



[그림 6.5] '확립'의 자음동화 처리에 의한 자연음과 합성음의 파형

본 논문에서는 운율 처리를 위하여 음량과 합성 주파수를 제어하도록 하였다. 그림 6.6은 단어 '안녕'의 문장 부호에 따른 합성 파형 중 마지막 음절이다.



[그림 6.6] 문장부호 '?'(위)와 '.'(아래)에 따른 '안녕'의 합성 파형

6.2. 결론

본 논문에서는 호스트로 부터 수신한 문장을 음성으로 출력할 수 있는 시스템을 설계하고 구현 하였으며, 그 결과를 실험을 통하여 확인 하였다. 이로써 컴퓨터에서 처리한 정보의 결과를 직렬통신선로를 통하여 음성으로 출력할 수 있으므로 RS-232C 통신 방식을 사용하는 대형 컴퓨터의 음성 단말기로의 사용이 가능하게 되었으며 음성 자동 응답 시스템(ARS)의 음성 출력 장치로도 사용이 가능할 것이다.

시스템에 문장을 입력하고 합성된 음성을 들었을 때, 자연음과 비교하여 다음과 같은 문제점이 지적되었다.

- 음절간의 천이가 자연스럽지 않다.
- 음절간의 음량이 일정하지 않다.
- 받침글자의 합성음을 알아듣기 어렵다.
- 문장부호에 의한 운율 처리가 자연스럽지 않다.

이와 같은 문제점들은 음소 DB의 고급화, 운율 처리 부분 강화및 음절간의 휴지시간을 조절함으로써 해결할 수 있을 것이며 받침글자에 대한 모음 음소를 음소DB에 추가하고 본 논문에서 3단계로 처리한 음량및 합성주기를 다 단계로 하면 보다 개선된 합성음을 얻을 수 있을 것이다.

파형 부호화방식에 의하여 구성된 음소의 합성 시간을 제어할 수 있는 알고리즘을 이용한 운율 처리와 복모음 합성시 모음간의 천이 과정등은 앞으로 더 연구 되어야할 과제이다.

참 고 문 헌

1. Analog Devices, Inc., ADSP-2101 Cross-Software Manual, 1989.
2. Analog Devices, Inc., ADSP-2101/2101 User's Manual, 1990. 2.
3. Ferrel G., Stremmer, Introduction to Communication Systems-3rd ed., ADDISON-WESLEY PUBLISHING COMPANY, 1990.
4. Kim Yong-Kap, "A Rule-Based Text-to-Speech System for Korean", KOREA-JAPAN JOINT WORKSHOP PN ADVANCED TECHNOLOGY OF SPEECH RECOGNITION AND SYNTHESIS, 1991. 7.
5. M. Morris Mano, COMPUTER ENGINEERING Hardware Design, Prentice-Hall, 1988.
6. M. Morris Mano, COMPUTER SYSTEM ARCHITECTURE-2nd ed.
7. Sajjan G. shiva, COMPUTER DESIGN AND ARCHITECTURE, Little, Brown, 1985.
8. 과학기술처, 음성인식 기술개발에 관한 연구-최종연구보고서, pp 76-88, 1987. 3.
9. 권영철, 음소단위의 규칙에 의한 음성합성에 관한 연구, 전국대학교 석사학위 논문, 1987.
10. 권용경, 김만수, 새한글 맞춤법사전, 집현전, 1992. 8.
11. 김진우, 이기문, 이상억, 국어음운론, 학연사, 1992. 3.
13. 안승권, 이용근, 성광모, 최상대, "포맷트를 이용한 한국어 규칙합성," 국제잡음·진동심포지움 및 한국음향학회학술발표회 논문집, 1989. 10.
14. 안수길, 임운천, "단음절 합성과 무제한 합성 기술," 음성신호처리워크샵, 1988. 8.
15. 이계영, 이상범 "한국어 음운 변동 처리를 위한 효율적인 Rule Base System의 구성," 전자공학회논문집 제28권 B편 제12호, 대한전자공학회, 1991. 12.
16. 이용주 "음성 합성 기술의 현황과 과제," KOREA-JAPAN JOINT WORKSHOP PN ADVANCES TECHNOLOGY OF SPEECH RECOGNION AND SYNTHESIS, 1991. 7.
17. 정곤양, 한국어 음성합성을 위한 하드웨어및 제어프로그램 개발에 관한 연구, 숭전대학교 석사학위 논문, 1983. 1.
18. 정연찬, 한국어 음운론, 개문사, 1992. 2.
19. 한국전자통신연구소, Man-Machine Interface를 위한 음성처리연구-최종연구보고서, pp 3099-347, 1991. 1.

부록 1. 한국어 문장-음성 변환 시스템의 DIP-SW 사양

1. BAUD RATE의 설정

S3	S2	S1	BAUD RATE
ON	ON	ON	1200
ON	ON	OFF	1600
ON	OFF	ON	2400
ON	OFF	OFF	3600
OFF	ON	ON	4800
OFF	ON	OFF	7200
OFF	OFF	ON	9600
OFF	OFF	OFF	19200

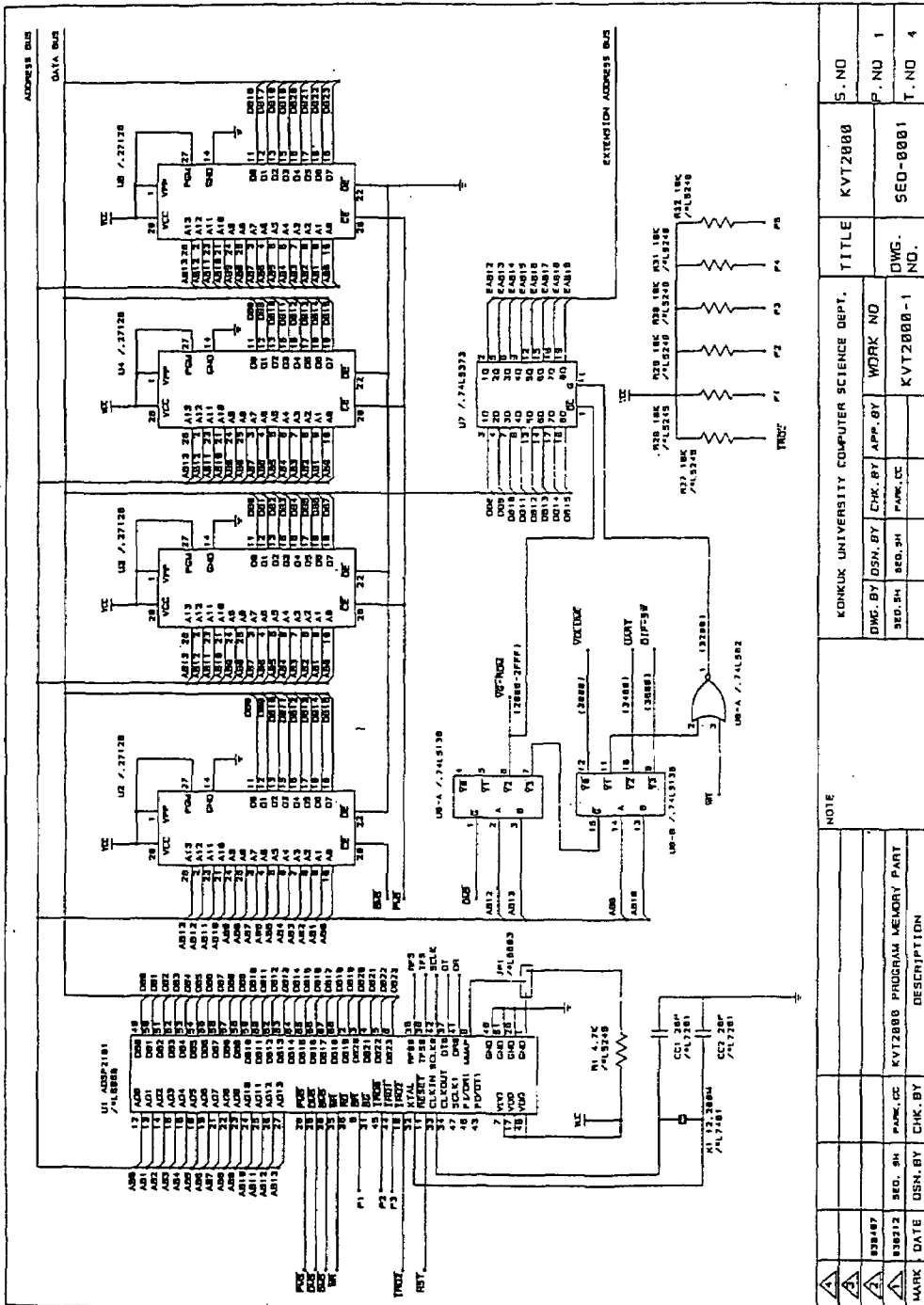
2. PARITY CHECK

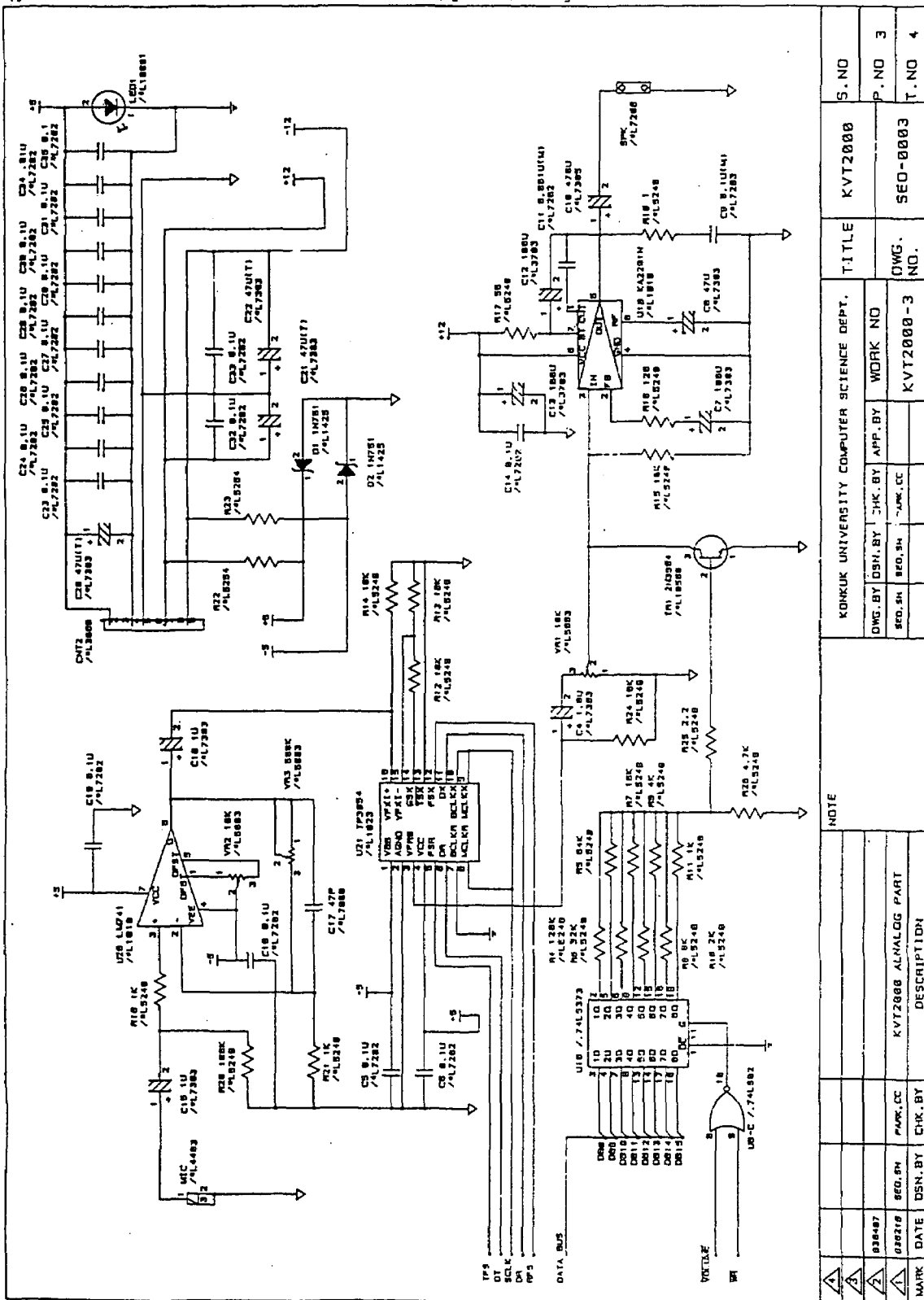
S5	S3	PARITY CHECK
ON	ON	NO PARITY
ON	OFF	ODD PARITY
OFF	ON	MARK PARITY
OFF	OFF	EVEN PARITY

3. STOP BITS

S6	STOP BITS
ON	1 STOP
OFF	2 STOP

부록 2. 한국어 문장-음성 변환 시스템의 회로도





.ADSP-2101 DATA MEMORY MAP				.KVT2000 MEMORY MAP				.VG-ROM EXTENDED ADDR			
<div>EXTERNAL RAM</div>				0000				0000			
				0400				0000			
<div>INTERNAL</div>				0800				2000			
				3000				EXTENSION			
				3400				3000			
				3800				3200			
				3C00				3400			
								3600			