

# 한글 統一符號에 관한 研究

## (Research on the Standard Code for Korean Letters)

金在均 \* ; 沈英錫 \* ; 金南哲 \*\*

(Kim, Jae Kyoon, Shim, Young Serk and Kim, Nam Chul)

### 要 約

한글 機械化에 수반되어야 할 한글 統一符號 制定에 관한 기본적인 문제들이 검토되었다. 한글의 특성을 충분히 살리기 위한 符號로서 모아쓰기 多重符號(multi-code) 방식을 제안하고 그에 준하여 computer 내부符號 및 telex 符號를 정하였다. 多重符號 방식은 符號 간에 퍼져있는 한글 情報를 符號自體 및 位置에 配定함으로써 端末에서의 display에 관한 情報를 쉽게 추출할 수 있고, 符號空間(code space)의 효율적인 활용에 의해 傳送速度에 있어서 풀어쓰기 방식에 비하여 7.6%의 증대를 얻을 수 있었다.

### Abstract

In this paper, some fundamental problems in the assignment and standardization of codes for Korean letters are discussed, and a multi-code method is proposed for the code assignment. This method is designed in particular to reflect the characteristics of Korean letters. Also, based on this method, a computer internal code and a telex code are suggested.

The multi-code method enables us to obtain information on the display at a terminal rather easily by assigning the information dispersed among the code sequences to code himself and position. Further this method is shown to be capable of improving the transmission rate by 7.6% over the conventional resolution method if the given code space is efficiently utilized.

### 1. 序 論

산업발달과 경제성장으로 情報社會가 되어감에 따라 情報의 交換 및 處理量이 급증하고 computer의 대중화로 인하여 한글의 機械化는 절실히 요구되고 있다. 따라서, 이에 수반되는 한글의 符號化는 매우 중요한 課題이며 이를 위한 종합적이고도 다각적인 연구

\* 正會員, \*\* 準會員, 韓國科學院 電氣 및 電子工學科

(Dept. of Electrical Science, KAIS)

接受日字 ; 1979年 5月 28日

가 필요하다.

한글의 機械化에 따라 현재 제정되어 쓰이고 있는 符號로는 한글用 ISO (International Organization for Standardization) code, telex code 등의 각종 情報交換用 符號가 있고,<sup>[1]</sup> 또 機種이 다른 computer마다 내부의 한글符號가 다양하여 사용자에게 큰 혼란을 일으킬 뿐더러 相互 情報交換에도 복잡한 符號變換 등의 불편이 따르고 있다. 그리고, 같은 computer 내에서도 sorting, display, print 등의 목적에 따른 符號變換이 필요하게 되므로 매우 번거롭다.

이러한 현상의 근본적인 이유는 한글은 24字的 基本字素들이 적절히 결합되어 하나의 文字를 구성함으로써 같은 表音文字이지만 하나의 基本자소가 하나의 문자를 나타내는 로마문자와는 다른 특성을 가지고 있어서 符號化에 많은 곤란이 따르기 때문이다.

즉 한글은 字素의 模樣과 位置가 基本자소 상호간의 관계에 의해 결정되므로 많은 情報量이 基本자소의 sequence에 퍼져 있어서 符號化時에 이들을 고려하지 않으면 한글 情報의 추출이 필요한 情報의 檢索 및 處理가 힘들게 된다. 따라서 한글의 符號化는 이러한 한글의 特性을 충분히 고려한 符號를 요구하게 되며, 또 그 統一化가 요구된다.

이러한 必要性에 따라 政府에서도 1974년에 5bix telex 부호, 7bit ISO 부호 등의 KS 부호를 정한 바 있으나, 이 부호들은 制定基準에 있어서 서로 關聯이 적어 相互變換이 힘들고 한글의 特性을 충분히 고려하지 못한 관계로 한글 機械化와 이에 따르는 각종 符號의 統一化 面에서 많은 問題를 안고 있다.

본 논문에서는 前述한 바의 문제들을 해결하는 方案으로 모아쓰기 多重符號 方式을 제안하고 그에 따라 符號配定을 하였다. 多重符號 方式은 한글의 한 문자가 初聲, 中聲, 終聲의 集合으로만 이루어짐을 이용하여 初聲, 中聲, 終聲을 같은 符號空間 內에 重複하여 配定함으로써 한정된 符號空間을 최대한 활용할 수 있다. 따라서 한글의 特性을 고려한 符號配定에 필요한 符號空間을 확보할 수 있게 된다. 이에 의하면 한글字素의 位置나 模樣에 대한 情報를 符號의 位置나 符號自體에 밀어 넣음으로써 情報檢索 및 端末 display를 간편히 할 수 있게 되며 나아가 初聲, 終聲의 情報를 符號自體에 넘겨 주면 情報 傳送速度의 增大를 기할 수 있게 되는 장점이 있다. 이와 같은 符號化 方式은 細密分離型 符號라 하여 端末에서의 display에 한해 이용된 바 있다.<sup>[2]</sup>

2. 한글 符號化에 對한 考察

한글 統一符號를 制定하는 데 관련된 일반적인 問題로서 入力字素의 數, 鍵盤과의 관계 및 符號의 形態, 그리고 國際 情報交換用 標準符號(ISO code)와의 관계를 서술한다.

2-1. 한글 入力字素의 數

실제 사용되고 있는 入力字素의 數는 여러가지이지만 풀어 쓴 入力形態는 표1과 같이 세 가지 類型으로 분류할 수 있다. 類型 A의 경우는 入力字素의 數는 매우 적으나 모아쓰기(formation)를 위해서는 每音節마다 delimiter(단음절 space)를 넣어야 하고,

類型 B는 類型 A의 경우에 雙字音 5字를 첨가하여 delimiter 없이 모아쓰기가 가능하도록 한 것이며, 類型 C는 雙字音 5字 대신 雙字音을 표시하는 parameter Δ를 넣어서 入力字素의 數를 줄인 것이다. 이외에 現用 3, 4번식 typewriter의 경우처럼 모아쓰기 형태의 入力도 생각할 수 있다.

표 1. 入力 한글자소의 類型  
Table 1. Type of Korean input character.

類 型	A	B	C
字素數	24	29	25
內 容	기본자모 24字 (ㄱ, ㅋ, ㆁ, ㆁ)	기본자모 24字, 쌍자음 5자 (ㄱ, ㅋ, ㆁ, ㆁ)	기본자모 24字, △ parameter (ㄱ, ㅋ, ㆁ, ㆁ)

2-2. 한글符號의 形態 및 鍵盤과의 關係

入力字素에 따른 出力符號는 모아쓰기型 符號와 풀어쓰기(resolution)型 符號로 나눌 수 있다. 여기서 풀어쓰기型이라 함은 符號空間 內에 配定된 字素의 數가 한글 端末裝置의 字素 數보다 적어 端末을 위해 다시 모아쓰기를 해야 하는 形態의 符號를 말하며, 모아쓰기型은 符號空間 內에 配定된 字素의 數가 한글 端末裝置의 字素 數와 같은 경우를 말한다. 기존 符號 system은 KS 7bit ISO 符號나 EBCIDIC의 경우와 같이 符號空間이 큰 경우를 제외하고는 모두 풀어쓰기型의 符號이다.<sup>[1]</sup>

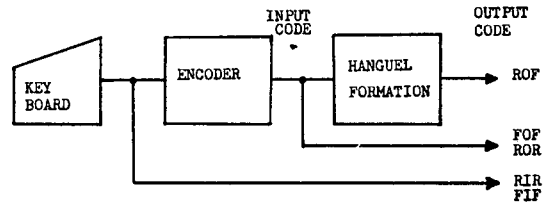


그림 1. 符號와 鍵盤과의 관계  
Fig.1. The relation between the code and key board.

出力符號의 發生과 鍵盤 사이의 관계를 圖示하면 그림 1과 같다. 이 때 출력符號의 發生構造는 풀어쓰기 入力-풀어쓰기 出力(RIR, ROR), 풀어쓰기 入力-모아쓰기 出力(ROF), 그리고 모아쓰기 入力-모아쓰기 出力(FIF, FOF)의 세 가지 類型으로 大別할 수

있다. 여기서 I, O는 出力符號의 發生位置를 encoder의 전·후에 따라 區分한 것이다. 그림 1에 圖示된 出力符號의 發生構造를 比較, 檢討하면 다음과 같다.

(1) RIR, FIF 型

RIR, FIF 型은 出力符號의 發生이 전반의 配列과 밀접한 關係를 가지고 있어서 한글이 아닌 他 文字와 혼용할 경우 편리한 鍵盤配列과 한글의 特性을 살린 符號配列의 두 가지 조건에 相互 背反이 일어나게 된다.

(2) ROR 型

ROR 型은 鍵盤과 獨立인 符號配定이 가능하나 풀어쓰기 符號이므로 한 音節이 차지하는 符號의 數가 2~7개로 變化하게 된다. 따라서 情報檢索, 또는 端末出力을 위해서 다시 모아쓰기를 해야 하는 불편이 따르게 된다.

(3) FIF, FOF 型

FIF, FOF 型은 入力字素의 數가 너무 많아 打鍵이 불편하고 영·한 혼용 時의 鍵盤配列이 거의 불가능하게 된다.

(4) ROF 型

그러나 ROF 型은 入力は 풀어쓰기로, 出力은 모아쓰기로 함으로써 한글의 特性을 충분히 살릴 수 있게 되며 출력符號의 發生이 鍵盤配列과 독립적이어서 打鍵에 편리한 鍵盤配列이 가능하다.

2-3. 國際 情報交換用 標準符號(ISO code)와의 關係

영·한 혼용 ISO 7 bit 符號를 制定하는 경우 그림 2-a와 같이 비어 있는 符號空間과 영어의 小文字

		$b_7 b_6 b_5$						
$b_4 b_3 b_2 b_1$	0	1	2	3	4	5	6	7
	control, graphics				영문 alphabet (대문자)			소 글 자 소

2-a. 영문용 소문자 부분 사용 時

		$b_8 = 0$							$b_8 = 1$							
		$b_7 b_6 b_5$ SI							SO							
$b_4 b_3 b_2 b_1$	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
	control, graphics				영문 alphabet				한글용 부호				한글 자소			

2-b. SI/SO key 사용 時 또는 8 bit code

그림 2. 한글용 7 bit ISO 부호 사용법

Fig. 2. The use of 7 bit ISO code for Korean letters.

에 配定된 符號空間을 이용하여 한글符號를 配定하는 방법이 있고, 그림 2-b와 같이 SO/SI(shift out/shift in) key를 이용하여 영문용 符號와 한글용 符號를 구분하거나 한 bit ( $b_8$ )을 영문과 한글용 符號를 구분하는 데 쓰는 8 bit 符號配定이 있을 수 있다. 그림 2-a와 같은 符號配定은 좁은 符號空間으로 인해 적절한 符號配定이 힘들고, 현재의 추세가 영어의 소문자와 control code를 더욱 필요로 하고 있어 英文符號와 重複이 되는 것은 바람직하지 못하다. 특히 國際 情報交換에 있어 많은 不便이 따르게 되므로 SO key를 이용하거나 8 bit 符號를 만들어 쓰는 것이 적절한 것으로 判斷된다.

3. 한글 統一符號의 制定基準

한글의 特性을 살리는 符號制定에 앞서서 우선 符號制定에 관한 基準과 그 順位가 주어져야 한다. 본 論文에서는 특히 중요한 몇 가지 問題들을 추출하여 아래와 같은 制定基準을 만들었다.

(1) 한글 統一符號는 ISO 7 bit code와 重複되지 않아야 한다.

영어는 국제적으로 共用되고 있으므로 한글 符號는 ISO code와 함께 사용하되 서로 重複되지 않도록 SO key를 이용하여 0~1 列에는 control code, 2~3

列에는 特殊記號, 4~7 列에는 한글字素를 配定하도록 한다.

(2) 情報檢索에 관한 고려

한글符號의 配列이 關係적인 辭典式 配列을 따르도록 하고 한글 한 音節이 차지하는 符號의 數를 固定시켜 computer 內에서의 sorting, searching 등의 情報檢索이 편리하도록 한다.

(3) 한글 端末裝置의 簡素化

한글의 端末裝置, 例를 들면 display terminal, line printer 등을 위해 初聲과 中聲, 終聲의 分類, 母音의 水平과 垂直, 長·短의 分類, 그리고 子音의 形態 決定이 쉽게 이루어지도록 하여 그 system 構造를 簡素化할 수 있도록 한다.

(4) 符號變換의 容易

크기가 다른 符號, 즉 5 bit, 6 bit telex 符號와 computer 內部符號와의 相互變換이 쉽게 이루어지도록 配定하여 情報交換이 편리하도록 한다.

(5) 情報의 傳送

情報傳送到 있어서 速度를 빠르게 하고 誤差率이 작도록 한다.

4. 한글 統一符號의 制定

4-1. 制定 例

앞 節에서 行한 한글 符號化에 대한 考察과 한글 統一符號의 制定基準에 바탕을 두어 아래와 같은 순서에 의해서 한글符號를 制定하였다.

(1) 한글용 7 bit ISO 符號는 SO/SI key를 사용하여 제정기준 1을 만족시키도록 하며 英文字素와 같이 4개의 列에 配定한다.

(2) 한글용 7 bit ISO 符號는 情報檢索과 端末에서 的 display에 편리하도록 모아쓰기 體으로 하고 符號配列은 辭典式 配列에 따르도록 한다.

(3) 限定된 符號空間을 최대한 活用하여 display에 필요한 한글字素의 位置와 模樣에 관한 情報를 추출하기 쉽도록 한다. 이러한 목적에 부응하는 符號化方式이 多重符號 方式이다. 이는 그림 3과 같이 모아쓰기를 할 때 初聲, 中聲, 終聲을 分類하여 각각에 대하여 같은 符號空間에 重複하여 符號配定을 하는 것을 말한다. 이렇게 함으로써 제한된 符號空間을 최대한 활용할 수 있고 初·中·終聲에 對한 情報 및 字素의 模樣에 관한 情報抽出이 쉽게 이루어진다.

(4) ISO 符號의 control character SP, DEL에 해당되는 곳은 ISO SO/SI key의 사용법에 따른 規約에 의해 符號配定을 하지 않는다.<sup>[3]</sup>

(5) 中聲 21字는 文字의 構造 및 形態를 규정짓는

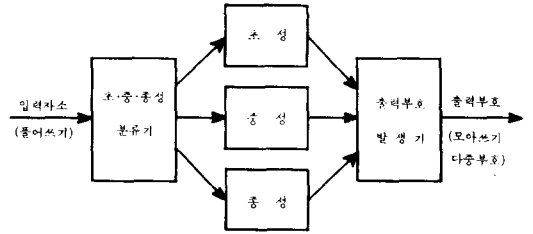


그림 3. 多重符號 발생 블록도  
Fig. 3. Block diagram of multi-code generation.

중요한 字素로서 水平, 垂直, 水平·垂直母音의 종류에 따라 한글의 글자모양이 달라지므로 이들은 分類하기 쉽도록 配定한다. 즉 수평모음은 0~3 行에, 수평·수직모음은 4~7 行에, 그리고 수직모음은 8~15 行에 配定한다.

(6) 6 bit telex 符號나 한글용 7 bit ISO 符號의 경우, 情報 傳送速度의 曾大를 위해 初聲, 終聲이 서로 겹치지 않도록 初聲 19 字는 4, 5 列에 終聲 26 字(無발침 기호 ㄱ 포함)는 6, 7 列에 配定한다.

(7) 한글용 7 bit ISO 符號와 5 bit, 6 bit telex 符號 사이의 相互變換이 쉽도록 한다. 즉 7 bit ISO 符號  $b_5 \sim b_1$ 은 5 bit telex 符號와 같게 하고  $b_6 \sim b_1$ 은 6 bit 符號가 되도록 한다. 그리고 5 bit telex 符號의 配定에 있어서는 오차를 생각하여 SP, LT, FI의 control character 部分에는 多重符號化를 하지 않는다.

표 2.-a. 5 bit 한글용 telex 부호  
Table 2.-a. 5 bit telex code for Korean letters.

$b_4 b_3 b_2 b_1 \rightarrow$	초 성		중 성		종 성	
	0	1	0	1	0	1
0000	KL				ㄴ	ㄹ
0001	KL				ㄷ	ㄹ
0010	LF	ㄹ	ㄷ	ㄴ	ㄱ	ㄹ
0011	ㄱ	ㄹ	ㄷ	ㄴ	ㄱ	ㄹ
0100	SP	ㅅ	ㅅ	ㅅ	ㅅ	ㅅ
0101		ㅅ	ㅅ	ㅅ	ㅅ	ㅅ
0110	ㄴ	ㅇ	ㄷ	ㄷ	ㄴ	ㅇ
0111		ㅅ	ㅅ	ㅅ	ㅅ	ㅅ
1000		ㅅ	ㅅ	ㅅ	ㅅ	ㅅ
1001	ㄷ	ㅅ	ㅅ	ㅅ	ㅅ	ㅅ
1010	ㄷ	ㅅ	ㅅ	ㅅ	ㅅ	ㅅ
1011	ㄷ	FI	ㅅ	ㅅ	ㅅ	ㅅ
1100		E	ㅅ	ㅅ	ㅅ	ㅅ
1101		ㅅ	ㅅ	ㅅ	ㅅ	ㅅ
1110		중	ㅅ	ㅅ	ㅅ	ㅅ
1111		LT	ㅅ	ㅅ	ㅅ	ㅅ

표 2-b 한글용 7 bit ISO 부호  
Table 2-b 7 bit ISO code for Korean letters.

		초 성		중 성			종 성	
b <sub>7</sub> b <sub>6</sub>		10		10		11		
column		4	5	4	5	6	7	
b <sub>4</sub> b <sub>3</sub> b <sub>2</sub> b <sub>1</sub>	b <sub>5</sub>	0	1	0	1	0	1	
		0000					ㄱ	ㄴ
	0001	ㄱ	ㄴ			ㄷ	ㄹ	
	0010					ㅁ	ㅂ	
	0011	ㄱ	ㄴ			ㅅ	ㅇ	
	0100					ㅈ	ㅊ	
	0101					ㅋ	ㆁ	
	0110	ㄴ				ㄷ	ㄹ	
	0111					ㅁ	ㅂ	
	1000					ㅅ	ㅇ	
	1001	ㄱ				ㅈ	ㅊ	
	1010	ㄱ	ㄴ			ㅅ	ㅇ	
	1011	ㄴ				ㅈ	ㅊ	
	1100					ㅋ	ㆁ	
	1101	ㄱ				ㅁ	ㅂ	
	1110	ㄴ				ㅅ	ㅇ	
	1111					ㅈ	ㅊ	

(8) 크기(bit 수)가 다른 code set 들을 다음과 같이 사용하기로 한다. 즉 5 bit telex 에 있어서의 情報傳送을 위한 5 bit telex code 는 computer 내에서 情報를 저장해야 할 경우 memory 절감을 위해서 사용할 수 있다. 7 bit ISO code 는 情報傳送 시 error detection 을 위해 parity bit 를 붙여 전송한다. 그리고 computer 내부에서 SI/SO code 를 포함하는 7 bit ISO code 를 쓸 수도 있으나 control code 가 더욱 필요하거나 file searching 이 쉽게 되어야 할 경우에는 parity bit 를 없애고 EBCDIC 과 같은 8 bit code 로 [1] 만들어 사용한다.

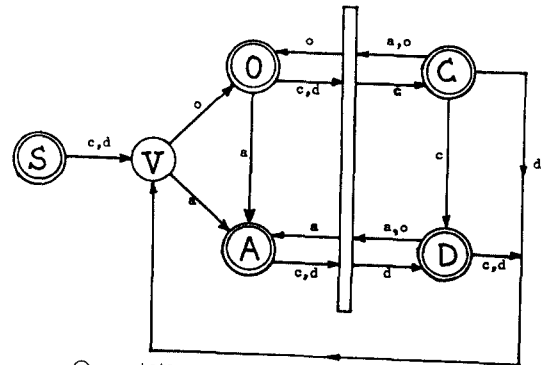
위의 같은 고려에 의해 制定된 한글용 5 bit telex 符號 및 7 bit ISO 符號는 표 2 와 같다. 여기서 初聲은 基本子音 14 字와 雙子音 5 字의 19 字에 對하여, 終聲은 基本子音 14 字와 雙子音 2 字(ㄱ, ㄴ)에 複合子音 9 字, 無받침 기호 ㄴ을 포함하여 모두 26 字에 對하여 符號化 하였으며, 中聲은 基本母音 10 字에 複合母音 11 字를 더하여 21 字에 對하여 符號化 하였다. 符號化한 복합자음은 그 出現頻度數가 많은 것부터 順序대로 택하였다.

4-2. 符號發生 및 모아쓰기 入力符號의 形態

出力符號를 얻기 위해서는 한글 모아쓰기를 하는 裝置가 있어야 하며, 또 鍵盤配列과 出力符號의 독립을 위한 encoder 가 필요하게 된다. 여기서 encoder 의 出力符號를 모아쓰기 入力符號라 한다.

入力符號 pattern 은 표 1 의 어느 類型으로 택하든

지 모아쓰기 出力符號 pattern 과 관계 없으나 打鍵하기에 편리하고 모아쓰기 裝置가 簡素化 되도록 2~2.5 벌식의 풀어쓰기 入力으로 한다. 이 때 鍵盤配列은 국제 표준 keyboard 上에 打鍵에 편리한 대로 한글 入力字素를 配列하되 모자라는 key 는 추가로 鍵盤위에 key 를 만들어 배치하여 現用 방식대로 특수문자, 영문, 한글의 세 가지 shift key 의 조작으로 入力되도록 한다. 구체적인 자판상의 key 배열과 入力 pattern 의 결정에 있어서는 앞으로도 많은 연구가 있어야 하겠다. 본 논문에서는 표 1 의 類型 B 와 같이 基本子音 24 字에 雙子音 5 字, 二重母音 4 字(ㅈ, ㅊ, ㅋ, ㆁ)를 첨가하여 入力字素의 數를 33 字로 하는 것이 打鍵하기에 매우 편리한 것으로 판단하여 이에 對해서 入力符號를 정하였다. 33 字 중 子音 19 字, 母音 14 字(C<sub>19</sub>, V<sub>14</sub>)를 入力字素로 하는 경우 모아쓰기를 위한 finite automaton 은 그림 4 와 같다.[4]



- state
- ◎ possible final state
- c 기본자음
- d 쌍 자 음
- a 수직모음
- o 수평모음

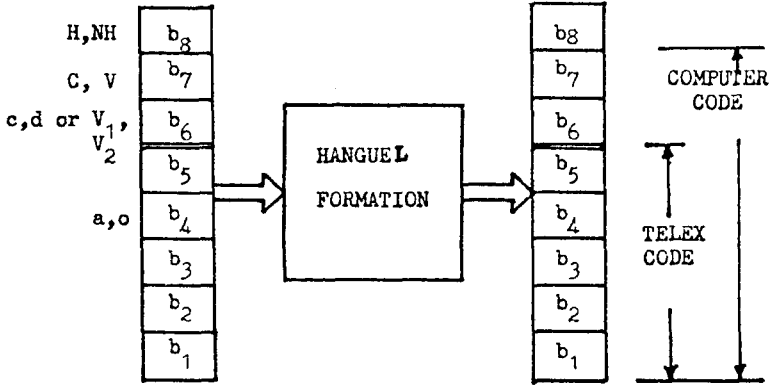
그림 4. C<sub>19</sub>, V<sub>14</sub> 에 대한 한글 모아쓰기 finite automaton

Fig. 4. Finite automaton for C<sub>19</sub>, V<sub>14</sub> Hangeul formation.

이 automaton 에서 모아쓰기와 出力符號의 發生을 쉽게 하기 위해서는 automaton 의 動作에 필요한 a, o, c, d 의 분류를 간편하게 하고 入力字素에 해당하는 出力符號에 관한 情報를 담고 있는 入力符號가 필요하다. 다음과 같이 入力符號를 만들면 위의 條件을 만족시킬 수 있다. 모아쓰기 入力符號를 8 bit 로 할 경우 b<sub>8</sub> 은 한글, 非한글, b<sub>7</sub> 은 子音·母音을 구분하는 bit 로 쓰고, 母音인 경우는 b<sub>3</sub> 에 의해 a·o 를 子音인 경우는 b<sub>6</sub> 에 의해 c·d 를 判別하도록 하고,

나머지 bit는 入力字素의 出力符號와 갈게 모아쓰기 入力符號를 定한다. 이 때의 入力符號의 形態와 配列은 그림 5와 부록의 표 4와 같고, 이 경우에 대한 한글 모아쓰기 및 出力符號의 發生에 관한 flow chart는 부록의 그림 8과 표 5에 나타나 있다.

ROF 多重符號 system은 그림 6에서 보는 바와 같이 既存 符號 system에 비해서 전체 system 구조가 간단하다. 우선 情報交換 時 간단히 符號變換器 (code converter)에 의해서 符號變換이 이루어지며, 情報檢索이나 端末에서의 display를 위한 복잡한 符



$V_1 \{ \text{ㅏ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ} \}$   
 $V_2 \{ \text{ㅚ, ㅜ, ㅠ, ㅡ, ㅣ} \}$

그림 5.  $C_{19}, V_{14}$ 에 대한 入出力 符號의 構造  
 Fig. 5. Input-output code structure for  $C_{19}, V_{14}$

5. 既存 符號 system과의 比較

앞에서 制定한 符號 system과 既存 符號 system을 다음과 같은 면에서 比較해 보자.

(1) system 構造

號變換이 필요없게 된다. 또한 telex 符號나 com-puter 内部符號의 發生構造가 類似하여 鍵盤에서부터 한글 모아쓰기까지의 block이 동일하다.

(2) 傳送速度와 誤差率

調査된 바에 의하면 有받침 文字, 無받침 文字 및 word space의 확률빈도를 각각  $P, P(\emptyset), P(S)$ 라 하고 한글 한 音節 당 基本字素의 數를  $\langle \beta \rangle$ 라 할 때

$$P = 0.3337$$

$$P(\emptyset) = 0.4153$$

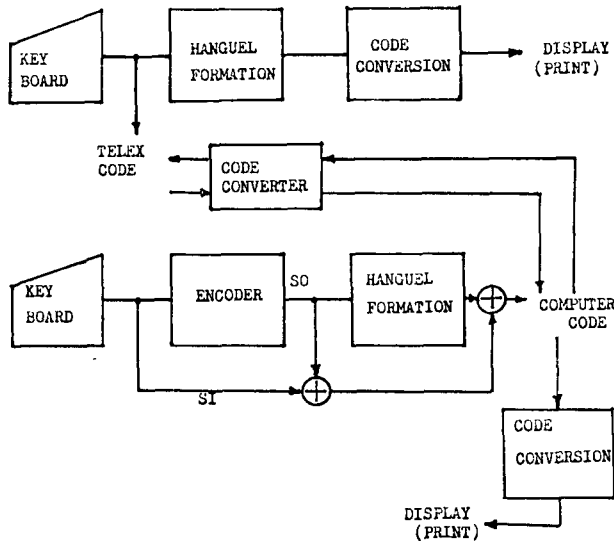


그림 6-a. 기존 부호 system의 블록도  
 Fig. 6-a. Block diagram of current code system.

$$P(S) = 0.2510$$

$$\langle \beta \rangle = 2.64 \text{ 기본자소/음절이다.}^{[5]}$$

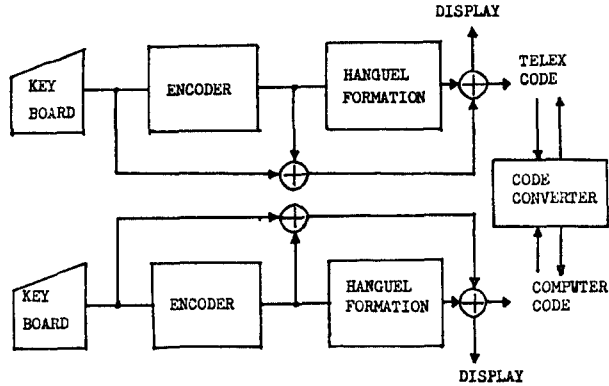


그림 6-b. ROF 多重符號 system의 블록도  
Fig.6-b. Block diagram of ROF multi-code system.

표 3. 음절당 부호수의 比較  
Table 3. Comparison of the number of codes per syllable.

부호의 종류		부호의 수/음절	system
모아쓰기 다중부호	1	2.44	6 bit telex 7 bit ISO
	2	3.00	5 bit telex
풀어쓰기	1	2.64	5 bit telex ( $\Delta$ parameter)
	2	3.39	5 bit telex (단음절 space)

이에 의해 전송속도의 尺度가 될 수 있는 한글 한 음節의 傳送에 필요한 符號의 數  $\langle \beta \rangle$ 를 구하면 표3과 같다. 이로써 정보검색을 위한 code  $\wedge$ 를 포함한 多重符號 방식과  $\Delta$  parameter를 이용한 풀어쓰기 방식 사이의  $\langle \beta \rangle$ 를 다음과 같이 比較해 볼 수 있다. 우선, 6 bit 또는 7 bit ISO code의 경우  $\wedge$ code를 제거하여도 수신측에서 다시 初聲, 中聲, 終聲을 구별해 낼 수 있도록 표2-b에서 보는 바와 같이 初聲과 終聲은 서로 다른 column에 配定되어 있다. 따라서 傳送時 무받침 文字의 경우  $\wedge$ code를 제거하여 보내면 표3에서와 같이  $\langle \beta \rangle$ 가 2.44 부호/음절이 되어  $\Delta$  parameter를 이용한 풀어쓰기 방식보다  $\langle \beta \rangle$ 가 7.6% 增大된다. 5 bit telex code의 경우에는 있어서는, 多重符號 방식에서  $\wedge$ code를 제거하면 수신측에서 初·中·終의 분류가 불가능하여 반드시 한글 한 文字당  $\wedge$ code를 포함시켜 初·中·終의 순서로 보

내야 하므로  $\langle \beta \rangle$ 가 3.0 부호/음절이 되어 단 음절 space를 사용하는 기존 telex에 비해  $\langle \beta \rangle$ 가 10.7% 높지만  $\Delta$  parameter 방식에 비해서는 11.5% 정도 떨어진다. 그러나 5 bit telex code의 경우 이와 같이  $\Delta$  parameter 방식에 비해 효율이 저하되는 단점이 있는 반면에 모아쓰기 多重符號 방식으로 할 때 오는 여러가지 면에서의 장점이 더욱 크다는 것을 주지해야 하겠다.

傳送時의 誤差 發生에 대해 생각하면, 먼저 한글 字素의 符號가 다른 한글字素 符號로 바뀌었을 경우 多重符號에서는 誤差가 생겨도 初聲, 中聲, 終聲의 情報은 잃지 않으므로 한 字素만 誤差가 생기지만, 풀어쓰기 符號에서는 심하면 두 文字까지 誤差가 퍼질 수가 있다. 그러나 7 bit ISO 符號나 6 bit 符號의 경우 한글字素 符號가, 特殊記號나 SI 이외의 control 符號로 바뀌는 경우 풀어쓰기 符號에서는 1~2 文字만이 誤差로 나타나지만 多重符號에서는 SP가 나타날 때까지 平均 2 文字 정도가 誤差로 나타나게 된다. 반면에 5 bit telex 符號의 경우는 無받침 記號  $\wedge$ 가 KL과 같은 符號로 重複 配定되어 있기 때문에 誤差로 나타나는 文字數를 더 줄일 수 있게 된다. 또 위와 같이 他 control 符號로 바뀌는 경우는 多重符號에서는 中聲이나 終聲에서의 誤差는 無視할 수 있으므로 실제의 誤差 文字는 더욱 줄일 수 있다.

(3) 符號變換의 容易性

既存 符號 system에서는 符號變換時 복잡한 符號變換器가 필요하지만 본 論文의 多重符號를 이용한 system에서는 다음과 같이 간단하게 할 수 있다. 우선 5 bit 符號를 6 bit 또는 7 bit ISO 符號로 變換할 경우에는 初聲은  $b_7b_6 = 10$ , 終聲은  $b_7b_6 = 11$ , 中聲은  $b_7b_6 = 1C_6$ 로 하고  $b_5 \sim b_1$ 은 5 bit 符號를 그대로 사용한다. 여기서,

$$C_6 = b_5 \oplus \bar{b}_4 + b_1(b_2b_5 + \bar{b}_3\bar{b}_4)이다.$$

그리고 7 bit ISO 符號를 5 bit 또는 6 bit 符號로

變換할 경우는  $b_7, b_6$  또는  $b_7$  bit 를 메어 버리면 된다.

(4) Display

한글을 display 할 경우 既存符號 system에서는 주어진 번식에 따른 情報를 추출하여 글자를 display 하거나, 무번식의 문자를<sup>[6]</sup> display 할 경우라도 初聲을 먼저 display 하고 뒤에 나오는 中聲의 種類, 終聲의 有·無에 따라 계속 模樣을 변화시켜 나간다. 이 때 필요한 情報의 抽出이 불편하여 端末裝置가 복잡하다. 그러나 본 論文의 符號를 채택하면 그림 7 과 같이 3 byte 를 보면 즉시 文字構造의 類型이 밝혀진다. 즉 初聲과 中聲의 크기를 받침의 有·無에 관한 情報는 終聲符號의  $b_5 \sim b_1$  의 5 bit 를 入力로 하는 AND gate의 出力에 의해, 그리고 初聲과 終聲의 模樣을 定하는 母音의 種類(水平, 垂直, 水平·垂直)에 관한 情報는 中聲符號의  $b_3 b_4$ 에 의해 결정된다. 따라서 그림 7 과 같이 memory 에 3 byte 로 한 音節을 저장하여 display 하면 端末 display가 간편해진다.

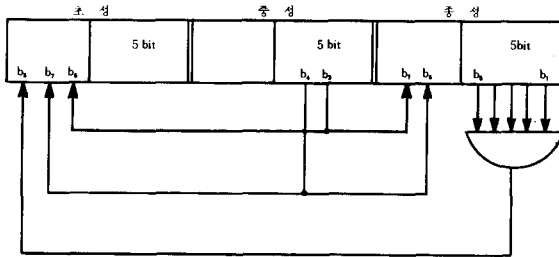


그림 7. 한글 display를 위한 data의 구조  
Fig. 7. Data pattern for Hanguel display.

(5) Memory size의 절감

한글 情報의 저장을 위해서 7 bit code를 표 2-a 와 같은 5 bit code로 변환시켜 한글 한 음절을 15 bit로 만들어 그대로 저장하거나 또는  $\wedge$  부호를 제거하여 variable length (10 bit 또는 15 bit)의 data를 저장하면 memory size를 크게 줄일 수 있다.

6. 結 論

多重符號 方式을 이용하여 符號空間을 충분히 확보할 수 있어서 한글 統一符號가 가져야 할 性質들을 상당히 實現할 수 있었다. 우선 전체 情報交換 system이 간단해지고 display 및 情報檢索이 편리하게 이루어지며, 크기가 다른 符號 사이의 相互變換도 쉽게 이루어진다. 앞으로 한글 機械化에 중요한 統一符號化 研究를 함에 있어서 먼저 制定基準에 관한 問題가

더욱 檢討되어야 할 것이며, 符號의 形態는 符號空間을 최대한 活用할 수 있고 音節 單位 符號의 意味가 있는 多重符號 方式이 適切할 것으로 보인다.

參 考 文 獻

1. 박 안 기 “중양전자계산연구소 1970년도 연구개발사업보고서,” 1970년 12월.
2. 안 수 길 “한글문자 모아쓰기 display의 한 방안,” 대한전자공학회지, 제 12 권 1 호, pp. 27~33, 1975년 2월.
3. CCITT “Data Transmission over the Telephone Network,” Orange Book vol. VIII, 1, pp. 18.
4. 최 광 무 “한글 모아쓰기에 관한 연구,” 한국과학원 석사논문 pp. 30, 1978.
5. 이 주 근 “韓國語 音節의 Entropy에 관한 연구,” 대한전자공학회지, 제 11 권 3 호, pp. 119-125, 1974년 3월.
6. 박 승 규 “컴퓨터에 의한 한글 기계화의 연구,” 한국과학원 석사논문, 1976년

\* 本 研究는 韓國科學院과 產學協同財團의 研究費 支援에 의하여 遂行된 것임.

附 錄

표 4. 한글 모아쓰기 및 출력부호 발생을 위한 입력부호

Table 4. Input code for Hanguel formation and output code generation.

$b_8$	1 (한글)							
	0 ( $V_1$ )				1 ( $V_2$ )			
$b_7$	0 ( $V_1$ )				1 ( $V_2$ )			
$b_6$	0 ( $V_1$ )		1 ( $V_2$ )		0 (c)		1 (d)	
$b_5$	0	1	0	1	0	1	0	1
$b_4 b_3 b_2 b_1$	0000							
	0001		..	π	γ	□		
	0010		..	..		ㅂ		



0011			ㅌ		ㄱ	ㅅ
0100					ㅋ	ㅆ
0101					ㆁ	ㅈ
0110			ㄴ	ㅇ	ㅊ	ㅉ
0111				ㅍ	ㅌ	ㅊ
1000	ㅌ					ㅆ
1001	ㅈ		ㅣ	ㅊ	ㅌ	ㅆ
1010	ㅊ			ㄴ	ㅋ	ㅆ
1011	ㅈ			ㄴ	ㅌ	ㅆ
1100	ㅊ				ㅌ	ㅆ
1101	ㅉ				ㅌ	ㅆ
1110	ㅊ				ㅌ	ㅆ
1111	ㅉ				ㅌ	ㅆ

표 5. 출력부호를 위한 routine  
Table 5. Routines for output code generation

	OUT DATA	$b_8$	$b_7$	$b_6$
OUT 1	input data ( $b_5 \sim b_1$ )	parity bit	1	0
OUT 2	STORE 1	"	"	1
OUT 3	STORE 1	"	"	$d_6$
OUT 4	STORE 1	"	"	1
	STORE 2	"	"	0
OUT 5	ㅌ	"	"	1
	STORE 1	"	"	0
OUT 6	STORE 1, 2 →복자음 부호	"	"	1
OUT 7	STORE 1, inpot data →복모음 부호	"	"	$d_6$

$d_6$ ; STORE 1 DATA의  $b_6$

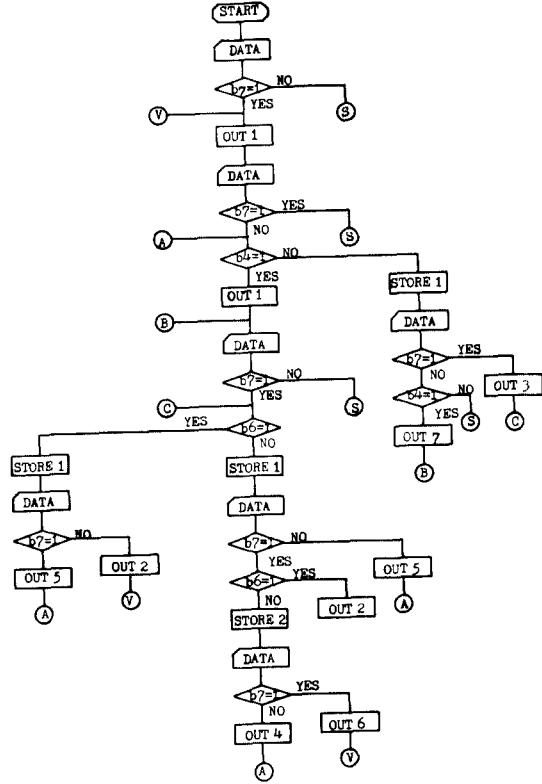


그림 8. 한글 모아쓰기 flow chart  
Fig. 8. Flow chart for Hangeul formation.