技術解說

電子計算機의 한글 入出力에 關한 硏究現況과 한글 반 풀어 쓰기 提案 (Ⅱ)

安 秀 桔*

8. 한글 모아 쓰기를 爲한 메카니즘

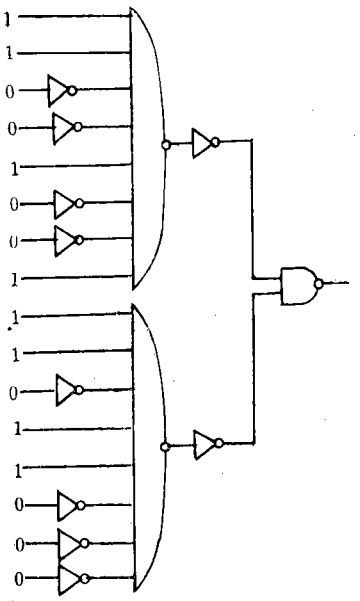
한글 字素 24個를 가지고 第6節에서 말한 바, 30가지 構成方式으로 組合해 가면 可能한 文字의 數가 11,000字를 超過한다(유시정, 참고문헌 7, p.36)고는 하지만, 印刷所에서 흔히 쓰는 活字數는 2,000個 程度이고, 공판인쇄기의 境遇도 2,000字를 넘지못 한다. 그 中 가장 複雜한 文字가 文字發生器의 最小의 크기를 決定지어 줄 것이니, "쐈"字의 境遇를 例로 삼는다면 절점수가 24, 枝路數가 18로서 Alphabet中 가장 複雜한 "E"文字의 6절점과 5枝路에 比해 거의 4倍임을 알 수 있다.

動員된 read only memory 素子가 클수록 收容可能한 圖案의 複雜性은 더 急히 增加하기 때문에, 한글 모아쓰기를 爲한 最大 記憶素子의 크기는 10×14 程度이면 足하리라고 추산된다.

하나의 64×7×5素子 크기의 文字發生器는 대강 16個의 모아쓰기 文字를 收容하게 된다는 計算이 되니, 공판 程度의 印刷나 display를 爲해서는 100個 程度의 文字發生器가 必要로 되는 것이다. 4個 程度의 上記 規模의 red only memory로서 모아쓰는 窮理가 없는 것은 아니나(안수길, 참고문헌 14) 原모아 쓰기의 red only memory의 必要經費는 文字發生器 單價가 $10 程度라면 全部해서 $1,000이 必要되는 것으로서 이것만으로서는 비싼 듯 하지 않으나, 함께 所要되는 全體材料만 해도, 特히 이러한 많은 ROM에서 올바른 address의 文字 무늬를 찾아 내가는 Demultiplexer만 해도 거의 不可能에 가까운 複雜性을 띄게 되는 것이다. 客談이지만, 人間 頭腦의 記憶容量의 구석구석에 숨은 情報를 적은 energy로써 찾아내는 能力에 다시 感嘆하게 된다. 한마디로 해서 한글 모아쓰기의 問題는 Alphabet의 境遇에 比해서 數十倍 乃至 數百倍 더 비싸게 먹힌다는 이야기이다. 이곳에서 數百倍로 잡는 것은, 素子가 많아져 가는 것에 비해 같은 screen上에 이 모든 文字를 表示시키는 데에 必要한 回路의 複雜性이 自乘에 比例하는 것으로 친 까닭이다.

電算機 內部에서 各各의 文字를 나타내는 內部 code에 關해서는, IBM과 UNIVAC에서는 8bit를 基本單位로 하고 있기 때문에 code의 個數가 많아 ($2^8=256$) 한글 字素에 對해서 別途로 code가 주어져 있으나, CDC에서는 內部 code가 6bit로 되어 있어서 Alphabet의 內部 code를 빌려 쓰고 있다. 한글의 境遇는 入力은 풀어쓰기로 하고 있지만, software的인 方法으로 모아쓰기로 變換해서 모아 쓴 한글 文字에 對한 內部 code를 주어야 한다. 그 具體的인 方案이 提案되었는데 (문명호, 참고문헌 13, p,77), 收容할 수 있는 한글의 文字數는 4,725이고 必要 bit數는 16이다. 이 內部 code를 判別하여 어느 文字인가를 찾아내는 (decoding) 過程에서, 材料의 個數가 역시 相當히 많아서 한 code에 對해서 8 input NAND TTL gate가 2個, inverter가 平均 10個로서 (第5圖) 1,500字만 decode해도 8 input NAND gate가 3,000個, hex-inverter가 2,500個, quad 2 input NAND gate가 約 400個가 드는데, 이러한 TTL IC 한 個에 $1로 計算해도 約 $6,000의 材料費가 들게 되는데 이것은 아직 decoding 段階, 다시 말해서 지금 電算機의 어느 한 adress의 2 byte memory에 어느 文字

* 正會員, 서울 大學校 工科大學



第5圖 16bit decoder의 一例 1100100111011000라는 신호가 들어올때만 신호를 내어 준다.

가 들어 있는가를 判定해 내는 데 쓰이는 것만 해서 그와 같이 든다는 것으로, 該當되는 文字發生器에서 文字圖形의 走査分解에서 나오는 binary 信號를 얻어 내어 CRT에 display 시키기 까지에는 아직도 먼 얘기이다. 上記 方法은 並列處理的 方法이고, 하나의 주어진 code에 對해서 1,500個 文字에 該當되는 code를 交代交代로 incoming code와 比較시켜서 判別을 行하는 方法도 있겠으나 最小限 1,500過程을 行하는 時間이 必要되기 때문에 system의 能率을 害치게 될 것이다.

한편 software的인 方法을 써서 內部 code와 incoming code와를 比較시키는 것도 있으나, 이 역시 直列動作이기 때문에 machine time이 消費되고, 그 때문에 結果되는 能率의 低下는 大型電算機일수록 심하다. 그런데 한글 問題를 다루기를 願하는 것은 大型機의 境遇가 많기도 하려니와 한글을 다룬다는 事實 自體가 大型機의 使用을 必要로 하게 만들기 때문에 能率의 低下는 더욱 深刻하게 된다.

가장 能率的인 software的인 解決方法은 display나 printing만을 爲해 小形電算機를 따로 쓰는 데에 있다. 近來 小形機의 價格低下($3,000~$4,000)로 보아 이것이 最善이 될 可望性이 있다. 24個 字母를 收容한 ROM에서 얻어진 直列信號를 CRT에 보내되 그 文字 中心(그곳을 中心으로 微細走査를 하는) 位置를 display 하고저 하는 文字에 따라, 다시 말해, 그것이 垂直母音을 가졌는지 또는 水平母音을 가졌는지 또는 받침이 있는지 없는지에 따라서 알맞게 移動시켜 주는 機能을 갖고 있는 "variable combination circuit"를 使用하여 24個 字素만을 갖고 모아 쓰기를 하는 案이 提示되었는데 (이주근, 참고문헌 10, p. 43), 이 境遇에 있어서 이 "variable combination circuit"은 display 하고저 하는 文字의 類型에 따라서 X 및 Y軸에 必要한 偏向을 주는 analogue 電壓을 digital 量으로부터 만들어 주는 D-A 變換器가 必要하고 (모든 row output character disply의 階段走査에서 이미 X軸, Y軸에 하나씩 必要한 것이지만), 이 X軸 및 Y軸 方向의 2個의 微細偏向用 D-A變換器의 clock pulse rate는 原走査用보다 훨씬 높아야 하면서 이 變換器들에 보내지는 位置 code는 文字에 따라 달라지기 때문에, 前述한 바 decoding 過程에서도 이미 상당히 複雜한 機構와 高價의 費用이 들게 되는 것이다. 따라서 모아쓴 한글 文字의 code 配當에는 初聲, 中聲, 終聲의 三部分을 code 길이를 역시 三分하여 各各 獨立的으로 配當하여 decoding도 全體 bit에 걸치지 않고 局部的으로 할 수 있게 해야 기구가 간단하게 될 것이다. 中心點을 옮기지 않는 範圍에서의 解決方案을 (안수길, loc. cit.) 찾아내었는데 不遠 發表할 豫定이다.

다시 line printer의 境遇를 考察하면, soft copy의 境遇와는 달라서 한꺼번에 display해 줄 必要는 없고, 印刷를 二段階나 三段階로 해서 各段階 사이에 많은 時間이 經過되더라도 最終的으로 모아 찍혀 고운 結果만 보여 주면 되니까 問題가 더 쉬운 듯 하지만, drum printer의 境遇나 chain printer의 境遇나 font 素子의 數가 많아질수록 印刷 速度가 늦어져서 問題가 된다. 特히 入出力 때문에 時間을 많이 빼앗기고 있는 것이 實情이고 보면 그것이 다시 더 느려진다는 것은 相當한 短點이라 하겠다. 그러나 하나의 한글

文字를 풀어 놓았을 때에는 하나의 字素를 찾아 가기는 比較的 빠르다 하드라도 二段階 三段階로 찍음으로 해서 特히 멀리 떨어져 있는 字素 사이를 왔다갔다 할 때에는, 오히려 chain이 길어져도 한꺼번에 모아 쓰는 경우가 比較的 덜 不利할 境遇도 있으리라는 것이 짐작되어서 複合字母를 많이 가진 IBM의 方式도 評價될 만 하다. 그러나 이미 말한 바와 같이 印刷된 글자가 적고 반칩이 두 가지 모자라는 點이 있으며, 그밖의 메이커— printer들은 글자가 크나 글자의 大小가 고르지 못하며 均衡이 안잡혀 곱지 못한 感이 있다. 이 境遇도 字素의 中心點을 옮겨줄 수 있다면(母音字素가 一字分 떨어지지 말고 自己左側에 있는 子音字素 바로 옆에 가서 찍히는 等) 좋지만, 그러한 line printer는 全혀 開發되어 있지도 않고, 開發된다손 치더라도 字母를 判定하여 中心點을 옮겨주는 機具가 複雜해서 單價가 올라갈 것이며, 또한 需要가 alphabet line printer에 比해서 너무 적기 때문에도 高價가 되는 흠이 있을 것이다.

結局 한글 모아쓰기는 歐文을 使用하는 各國에서는 想像못할 程度의 비싼 欲望이 되고마는 것이다.

2벌式 程度의 簡單한 한글打字機 字板으로 찍으면서도 4벌式 程度의 고운 出力을 내어주고 그러면서도 速度가 빠른 打字機 專用 超미니 콤퓨터—를 쓰는 方式이 硏究되었는데(朴憲緖, 참고문헌 15) 이는 글자가 고운 4벌式은 여러벌의 鍵中 어느 편을 찍어야 하는지 等 찍을 때 이미 判斷力을 動員해야 하는데 그 不便을 없애서 打字速度가 커지며 손쉬워지는 利點이 있으나 한글打字機用으로서 모아 쓴 한글의 一字分이 한꺼번에 찍히는 것일 수는 없고 電算機用 line printer를 위한 硏究는 아니나 console printer나 teletype에는 쓸 수 있을 것이다.

9. 한글 풀어 쓰기

一國의 言語에 關聯되는 限, 不可避한 事情에 直面하기 전에는 너무 變更을 시키지 말아야 하는 것이기 때문에, 그리고 電算機가 導入되고 活用된 時日이 짧기 때문에 跳躍的 發展을 할 素地를 안은 채 한글의 障壁을 뚫지 못한 狀態로 남아 있으나, 이 分野의 硏究者들은 한결같이 풀어쓰기를 提唱하고 있고 (성기수, 참고문헌 1, pp. 62~68), (이주근, 참고문헌 5, p. 10), (박안기, 참고문헌 p, 122), 나머지 參考文獻에서는 이미 풀어쓰기를 前提로 하여 硏究를 하고 있다.

한글을 풀어 썼을때, 10個 中聲字素中 垂直母音 ㅣ, ㅏ, ㅓ, ㅑ, ㅕ를 풀어 써도 (橫書의 境遇)形態는 그대로 있고, 子音과 母音字素間 距離만이 모아쓰기의 境遇에 比해 더 떨어져 있는 것이기 때문에 比較的 읽기가 쉽지만, 水平母音 ㅗ, ㅛ, ㅜ, ㅠ, ㅡ는 풀어쓰기를 했을 때 더 읽기가 어려워지는 것을 볼 수 있다. 第6圖에서 垂直母音이 많은 (a)의 境遇와 水平母音이 많은 (b)의 境遇를 比較해 보면 前者가 比較的 읽기가 쉽다는 것을 볼 수 있다. 이것은 垂直母音은 初聲과의 距離가 떨어질 따름이고 方向에는 變動이 없으나 水平母音의 경우는 두 字素 사이 距離도 멀어지지만 下部에 있던 母音이 옆으로 오는 데에서 더욱 읽기 힘들게 되는 까닭이다. 垂直母音의 경우는 比較的 보기가 容易한 편이므로 제대로 풀어쓰기를 하되 水平母音의 경우는 풀어 쓰지 말고 初聲과 中聲까지만 모아 쓰기로 한 것을 水平式 반 풀어쓰기라고 부르기로 한다면 完全 풀어쓰기와 水平式 半 풀어쓰기의 比較로 第7圖에서 해 볼 수가 있다. (a)가 完全 풀어쓰기이고 (b)가 水平式 풀어쓰기이다. 이때 반 풀어쓰기가 完全 풀어쓰기의 경우보다는 越等히 쉬운 것을 볼 수 있다.

(a) ㅇㅣㄹㅂㅏㄴㅈㅕㄱ ㅅㅏㅇㅌㅏㅣㄱㅏ
ㅇㅏㄴㅣㄷㅏ.
(b) ㅎㅡㅇㅂㅜㄹㅡㄹ ㄴㅗㄹㅂㅜㄱㅏ ㅉㅗㅊㄱㅗ.

第 6 圖

한편 反對로 세로 쓰기의 경우를 볼 것 같으면 終聲이 역시 距離만 떨어지고 位置는 自然스러운 下位에 있기 때문에 越等히 有利하나, 中聲의 位置는 도리혀 垂直母音이 不利하고 水平母音이 有利하다. 따라서 이번에는 水平母音을 떼어 쓰고 垂直母音을 붙여 쓰면 第8圖와 같이 된다. 但 水平型과 垂直型이 섞인 複母音의 경우 初聲에 水

(a) ㄱㅣㅁㅣ ㄷㅗㄱㄹㅣㅂ ㅅㅓㄴㅇㅓㄴㅁㅜㄴ
ㅇㅗㄷㅡㅇㅇㅡㄴ ㅈㅏㅇㅓㅣ ㅇㅏㅈㅗㅅㅓㄴㅇㅡㅣ ㄷㅗㄱㄹㅣㅂㄱㅜ ㄱㅇㅣㅁㄱㅗㅏ ㅈㅅㅓㄴㅇㅣㄴㅇㅡㅣ
ㅈㅏㅈㅜㅁㅣㄴㅇㅣㅁㅇㅡㄹ ㅅㅓㄴㅇㅓㄴㅎㅏㄴㅗㄹㅏㅊㅏㄹㅗㅅㅅㅓ ㅅㅓㅣㄱㅕㅣ ㅁㅏㄴㅂㅏㅇㅇㅓㅣ
ㄱㅗㅎㅏㅇㅕ ㅇㅣㄴㄹㅠ ㅍㅕㅇㄷㅡㅇㅇㅡㅣ ㄷㅏㅣㅇㅡㅣㄹㅡㄹ ㄱㅡㄱㅁㅕㅇㅎㅏㅁㅕ ㅊㅏㄹㅗㅅㅅㅓ
ㅈㅏㅅㅗㄴ ㅁㅏㄴㄷㅏㅣㅇㅓㅣ ㄱㅗㅎㅏㅇㅕ ㅁㅣㄴㅈㅗㄱ ㅈㅏㅈㅗㄴㅇㅡㅣ ㅈㅓㅇㄱㅜㅓㄴㅇㅡㄹㅇ
ㅕㅇㅇㅠㅋㅓㅣ ㅎㅏㄴㅗㄹㅏ

(b) ㄱㅣㅁㅣ ㄷㅗㄱㄹㅣㅂ ㅅㅓㄴㅇㅓㄴㅜㄴ
오드ㅇㅗㄴ ㅈㅏㅇㅔ ㅇㅏ ㅈㅅㅓㄴㅗㅣ ㄷㅗㄱㄹㅣㅂ구ㄱㅇㅣㅁㄱㅗㅏ ㅈㅗㅅㅓㄴㅇㅣㄴㅗㅣ ㅈㅏㅈㅜㅁㅣㄴㅇㅣㅁㅗㄹ
ㅅㅓㄴㅇㅓㄴㅎㅏㄴㅗㄹㅏ ㅊㅏㄹㅗㅆㅓㅅㅔㄱㅖ ㅁㅏㄴㅂㅏㅇㅇㅔ ㄱㅗㅎㅏㅇㅕ ㅇㅣㄴㅠㅍㅕㅇㄷㅗㅇㅗㅣ
ㄷㅐㅗㅣㄹㅡ ㄱㅡㄱㅁㅕㅇㅎㅏㅁㅕ ㅊㅏㄹㅗㅆㅓ ㅈㅏㅅㅗㄴ ㅁㅏㄴㄷㅐ ㅇㅔㄱㅛㅎㅏㅇㅕ ㅁㅣㄴㅈㅗㄱ ㅈㅏㅈㅗㄴㅗㅣ
ㅈㅓㅇㄱㅜㅓㄴㅗㄹ ㅇㅕㅇㅠㅋㅔ ㅎㅏㄴㅗㄹㅏ

第 7 圖

平型을 붙여 쓰면서도 垂直母音을 띄어 쓰는 (第7圖 (b)) 경우는 比較的 괜찮으나 그 反對는 어려우므로 ㅘ, ㅙ, ㅚ, ㅝ, ㅞ, ㅟ, ㅢ 等 7個 母音은 모아쓴 것이다. 이 두 가지 반 풀어쓰기의 경우와 完全 풀어쓰기의 경우 display 등을 위해서 우리가 具備해야할 素子 數는 完全 풀어쓰기의 경우는 24個이고 水平式반 풀어쓰기는 모아 쓰기를 위한 5가지 水平母音 ×19가지 初聲=95 풀어쓰기를 위한 初聲 19가지+ 終聲 複子音 10가지 其他 垂直母音 ㅏ, ㅑ, ㅓ, ㅕ, ㅣ, ㅐ, ㅔ, ㅒ, ㅖ 等 9가지를 合해서 133個이고 垂直式 반 풀어 쓰기는 (5가지 垂直母音+11가지 複母音)×初聲 19가지=304 및 풀어쓰기를 위한 19가지 初聲+ 10가지 終聲 複子音 其他 水平母音 ㅗ, ㅛ, ㅜ, ㅠ, ㅡ 5가지를 合해서 338個이다. 따라서 垂直式 반풀어쓰기와 水平式 반 풀어쓰기는 各各 完全 풀어쓰기에 比해서 素子數로만 해도 14倍와 5.5倍가 要求되나 line printer의 chain에 실을수 있는 範圍로 간주된다. 垂直式 풀어쓰기와 完全 풀어쓰기의 比較를 第9圖에서 해 볼 수 있다.

10. 結 論

打字機 字板과 同一한 配列로 key punch 한글 字板을 統一하는 것이 必要하고 時急하다. 그러나 한글 打字機 字板도 電子的 方法에 依한 判定 mechanism을 通해 4벌式 字板을 없애고 2벌式 十重子音 合計 29字素 程度로 入力시키고 出力은 4벌式 乃至 김동훈式의 活字로 적히게 하는 날이 멀지 않을 것이다.

line printer나 Display 裝置를 위해 생각해 보면 한글은 24字素 풀어 쓰기 (完全 풀어 쓰기)段階에서 이미 Alphabet보다 복잡하다. 이것은 平均 절점 數와 枝路 數에서도 나타나지만 最大 分解能을 要求하는 가장 複雜한 字素끼리의 比較 (ㅌ와ㅊ)에서도 같은 20% 程度의 增加를 볼 수 있다. 한 文字가 오직 初聲, 中聲, 終聲의 部分으로 되어 있는 것으로 생각해서 모든 複合字素를 Element로 생각하는 51字素의 경우 가장 複雜한 字素는 ㅙ으로서 約 2.3倍, 따라서 Alphabet를 爲한 5×7의 ROM 가지고는 不可能하다.

우리가 오늘날의 key punch 字板의 混亂도 國
ㅂㅏㄴㅁㅏㄴㄴㅓㄴ ㅇㅕㄱㅅㅏㅇㅡㅣ ㄱㅜㅓㄴㅇㅜㅣㄹㅡㄹ ㅈㅏㅇㅎㅏㅇㅕ ㅈㅏㄹㅡㄹ ㅅㅓㄴㅇㅓㄴㅎ
ㅏㅁㅇㅣㅁㅕ ㅇㅣㅊㅓㄴㅁㅏㄴ ㅁㅣㄴㅈㅜㅇㅇㅡㅣ ㅅㅓㅇㅊㅜㅇㅇㅡㄹ ㅎㅏㅂㅎㅏㅇㅕ ㅊㅏㄹㅡㄹㅍㅛㅛ
ㅁㅕㅇㅎㅏㅁㅇㅣㅁㅕ ㅁㅣㄴㅈㅗㄱㅇㅡㅣ ㅎㅏㅇㄱㅜ ㅇㅕㅇㅣㄹㅎㅏㄴ ㅈㅏㅇㅠㅂㅏㄹㅈㅓㄴㅇㅡㄹ
ㅇㅜㅣㅎㅏㅇㅕ ㅊㅏㄹㅡㄹ ㅈㅜㅈㅏㅇㅎㅏㅁㅇㅣㅁㅕ ㅇㅣㄴㄹㅠㅈㅓㄱ ㅇㅑㅇㅅㅣㅁㅇㅡㅣ ㅂㅏㄹㄹㅗ
ㅇㅓㅣ ㄱㅣㅇㅣㄴㅎㅏㄴ ㅅㅓㅣㄱㅕㅣ ㄱㅏㅣㅈㅗㅇㅡㅣ ㄷㅏㅣㄱㅣㅇㅜㄴㅇㅓㅣ ㅅㅜㄴㅇㅡㅇ ㅂㅕㅇ
ㅈㅣㄴㅎㅏㄱㅣ ㅇㅜㅣㅎㅏㅇㅕ ㅊㅏㄹㅡㄹ ㅈㅓㅣㄱㅣㅎㅏㅁㅇㅣㄴㅣ ㅅㅣ ㅊㅓㄴㅇㅡㅣ ㅁㅕㅇㅁㅕㅇ
ㅇㅣㅁㅕ

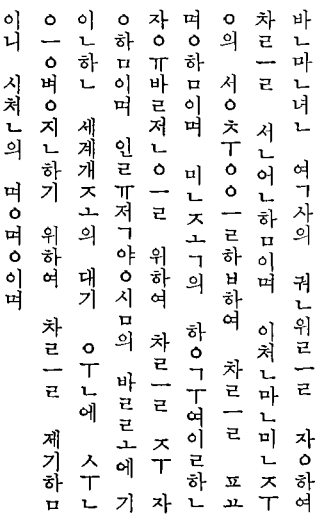
第 9 圖 (a)

家的인 對策이 앞서 있었더라면 避할 수 있었을 것이었다는 것을 생각할 때 풀어쓰기나 반 풀어쓰기도 國家的 scale에서 定해져야 되겠는 데 우리가 小形機의 경우 不便함을 무릅쓰고 Assembler language를 使用한 것과 마찬가지로 우리에게 가장 便하고 보기 좋다고는 하지만 모아쓰기를 固執하면 너무나 비싸게 먹히기 때문에 우선 너무 비싸지 않은 範圍에서 우리들의 감각도 尊重한다는 뜻으로 타협點을 찾아 반 풀어쓰기를 고려하여야 할 것이다.

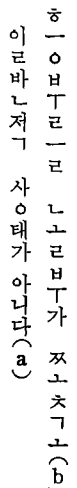
參 考 文 獻

(1. ~ 8.은 前號에 記載되어 있음)

9. 문교부 : "우리 말에 쓰인 글자의 찾기 조사" 1955년.
10. 이주근, 이균하 "조합 방식의 charater generator에 관한 연구" 1972년도 추계전자학술연구발표회 논문요약집 1972년 9월.
11. 차성부. "한글 문헌 정보 검색 씨스템 구성 연구" 성균관대학교 경영행정대학원 학위논문.
12. 이주근, "한글 문자의 인식에 관한 연구(Ⅳ)" 대한전자공학회지 제9권 제4호.
13. 문명호 "한글의 기계처리에 대한 고찰" 성균관대학교 경영행정대학원 학위논문 1971년 12월.
14. 안수길 "한글 문자 모아쓰기 display의 한 방안"미발표.
15. 박헌서 "초소형「한글 콤퓨티」개발" 동아일보 과학란 1973년 3월 9일자.



第 9 圖 (b)



第 8 圖