한글 코드에 무관한 한글 디코더의 설계에 관한 연구*

최 익 수°, 황희웅

서울대학교 컴퓨터 공학과

Design of Korean Decoder adaptable any korean code system

I.S. Choi, H.W. Hwang

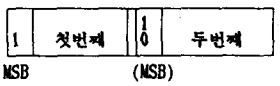
Seoul National University, Dept. of Computer Engineering

Abstract

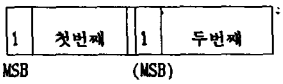
Many Korean graphic cards have been produced. But, due to the diversity of korean code system all korean graphic cards produced have been adapted in specific code system. Therfore, considering in KSC-5601-1987, the paper propose a Korean decoder independent of code system

1. 서론

컴퓨터에 대한 사용자 인터페이스로써 가장 많이 쓰이고 있는 것이 키보드와 모니터라고 할 수 있다. 그런데, 컴퓨터가 서양에서 개발되었기 때문에 영숫자만을 고려하였다. 따라서, 우리나라에서의 컴퓨터에 대한 인터페이스는 한글이 아닌 영문으로 이뤄지고 이를 개선하기 위해서 채택한 한글 카드에서는 많은 문제가 야기되었다.[4] 대표적인 문제점중에 첫째는 정보 처리 단위의 상이성인데 컴퓨터는 영숫자를 기준으로한 1 바이트 ASCII 코드 처리를 기본으로 하였기 때문에 2 바이트 코드 체계를 사용하는 한글로 사용자 인터페이스를 개발한다는 것은 상당히 어려운 일이다. 두번째로는, 현재 사용되고 있는 한글 코드의 난립이다. 영숫자의 경우에는 이미 ASCII 코드로 표준화 시켰고 우리와 상황이 비슷한 일본의 경우에도 표준화된 2 바이트 코드 체계를 사용하고 있으나 우리나라에서는 회사마다 다른 코드를 채택함으로써 타기종과의 호환성이 결여되고 있다는 것이다. 다행스럽게 1987년에 표준 연구소에서 2 바이트 완성형 코드 체계를 발표하였다.[1] 세번째로는, 영숫자의 경우에는 풀어쓰기 형태인데 반해 한글은 모아쓰기 형태이다. 따라서, 영숫자는 256(2^8)개의 코드로도 원하는 영숫자를 충분히 표시할 수 있는 반면 한글의 경우에는 초/중/종성에 대한 코드를 각각 배정하기 위해 필요한 최소 5 비트씩을 사용했을때 2 바이트의 조합된 코드를 생성하게 된다. 이러한 코드 체계의 장점은 한국어 음운학상 허용되는 모든 한글 음절이 생성될 수 있다는 것이다. 반면, 사용자가 많이 쓰고있는 글자 2350개에 대해서 조합된 형태로 코드를 제공하지 않고 순차적으로 코드를 배정함으로써 글자 모양을 좀더 미려하게한 2 바이트 완성형 코드가 있다. 이를 그림으로 나타내면 그림 1과 같다.[1]



(a) 2 바이트 조합형 코드



(b) 2 바이트 완성형 코드

그림 1. 한글 코드

이와같은 상황하에서 기존의 컴퓨터 시스템에 한글을 표시하기 위해서 조합형 코드와 완성형 코드중 하나를 선택하여 그 코드 체계에 맞게 그래픽 카드를 설계하였기 때문에 조합형 코드체계를 따르는 한글 카드는 완성형을, 완성형 코드체계를 따르는 한글 카드는 조합형 코드를 지원하지 못함으로써 발생하는 비효율성 문제에 부딪히게 되었다. 물론 기존에 시판되는 한글 카드중에는 조합형인지 완성형인지를 선택하여 선택된 코드 체계에 따라 다른 하드웨어를 통해서 한글을 표시하게하는 것도 나와 있지만 이는 사용자가 일일이 사용 코드 체계를 알려 주어야 한다는 불편이 있다. 위의 접근 방법의 또 한가지 문제점은, 특정 코드 체계로 쓴 화일은 사용자가 볼 때에 어떤 코드 체계를 따랐는지 알 수 없으므로 한번의 실패를 맛본 후에야 비로소 올바른 코드를 정의하여 한글 카드를 tunning할 수 있다는 것이다.

* 본 연구는 '87 목적 기초 연구의 일부로 수행되었음

따라서, 본 고에서는 위와 같은 문제점을 극복하여 어떤 코드 체계를 따르던지간에 올바른 한글을 표시할 수 있는 한글 디코더를 제안하고자 한다.

2. 시스템 개요

본 시스템은 IBM-PS/2용 그래픽 어댑터인 VGA에서 지원하는 소프트웨어와 호환성을 유지한 상태에서 한글을 표시하려고 하였기 때문에 전체 시스템 구성은 크게 3부분으로 나뉘어지는데 이를 블럭도로 표시하면 그림2와 같다.

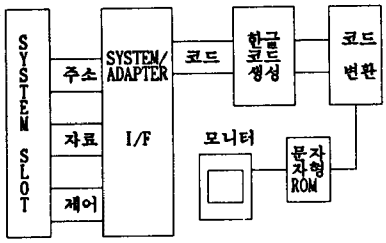


그림 2. 전체 시스템 구성

이중 인터페이스 부분은[6,7,8,] 모토로라사의 6845 S CRTC와 몇가지 부가회로로 구현하였고, 본 고에서 가장 중요시 하는 한글 코드 생성 부분과 코드 변환 부분을 중심으로 서술하겠다.

(1) 한글 코드 생성

한글 코드와 생성부는 원칙적으로 컴퓨터 시스템은 1 바이트 코드 처리 체계를 따르고 있으므로 2 바이트 코드 처리 체계를 따르는 한글의 경우에는 2 바이트 코드를 비디오 버퍼에서 읽어오는 과정이 필연적이다. 따라서, 비디오 버퍼에 있는 연속된 2 바이트 한글 코드-조합형이건 완성형이건 간에-를 적절히 래치하여 처리할 수 있도록 하는 메카니즘이 제공되어야 하는 바, 그의 블럭도는 그림3과 같다.

먼저, CRTC가 비디오 버퍼를 참조하기위한 주소를 생성하면 1 바이트의 자료가 그림 3의 래치와 버퍼에 실린다. 이때 첫번째 코드는 첫번째 참조시, 두번째 코드는 두번째 참조시 읽어 들일 수 있으므로 래치와 3-상태 버퍼에 위와 같이 클록과 enable신호를 주게 되면 비디오 버퍼에 대한 세번째 참조시에는 2 바이트의 한글 코드가 유효한 상태로 있게 된다.

(2) 코드 변환

본 고에서 채택하고 있는 코드 체계는 2-바이트 완성형으로서 해당 font ROM을 참조할 때에도 완성형

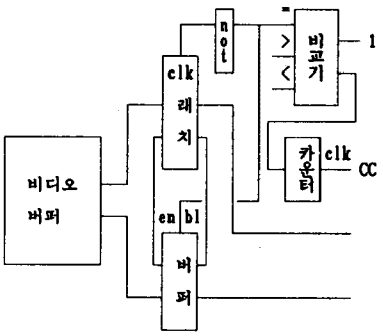


그림 3. 한글 코드 생성 논리

font를 가져와서 한글을 표시하게되므로 기본적으로 비디오 버퍼에 기억되어 있는 코드가 조합형일 경우에는 이를 적절한 완성형 코드로 변환시켜 font ROM에 참조해야만 한다. 그러나, 조합형 코드에서는 생성되나 완성형 코드의 제한성(한글의 경우 2350자)때문에 완성형 코드가 없는 부분이 있으므로 영역을 검사하는 부분이 있어야 한다. 또한, 비디오 버퍼에 있는 조합형/완성형 코드에 무관하게 완성형 font를 표시해야 하므로 그 변환 논리를 다음과 같은 철학하에서 설계하였다. 첫번째로, 해당 코드 변환 논리는 완성형 코드의 경우에는 bypass, 조합형의 경우에는 변환 논리를 통하여 완성형 코드로 변환 하게 하였고, 두번째로, 그림1에서 살펴보았듯이 2번째 바이트의 MSB가 완성형의 경우에는 항상 1인 반면 조합형의 경우 에는 0과 1의 두가지로 나뉘어져 2번째 바이트의 1인 것만으로는 조합형/완성형 코드를 구분하지 못하므로 조합형에서 2번째 바이트의 MSB를 1로 만드는 중성인 ㅐ, ㅑ, ㅒ, ㅘ, ㅙ, ㅚ, ㅞ, ㅟ, ㅢ는 한국어 문장상에서 계속하여 입계치-본 고에서는 10-만큼은 일어나지 않는다는 관찰 특성에 의해서 입계치만큼의 한글 문자 코드를 비디오 버퍼에서 읽어왔을때 만일 그 코드 체계가 완성형이었을 때에는 제대로 된 한글이 출력되어 코드 변환이 불필요하고 만일 조합형 코드 체계였다면 예기치 않은 글자가 출력되다가 입계치 이상이 되면 변환 논리를 거쳐 나온 완성형 코드에 의한 제대로된 한글이 출력된다. 이에 대한 블럭도는 그림4와 같다.

3. 결론

이제까지 본 고에서는 난립되고 있는 한글 코드 체계상에서 한국 표준연구소가 1987년에 표준안으로 내놓은 KSC-5601에 의한 2바이트 완성형 코드 체계를 궁극적으로 채택한 코드에 무관한 한글 디코더를 제안

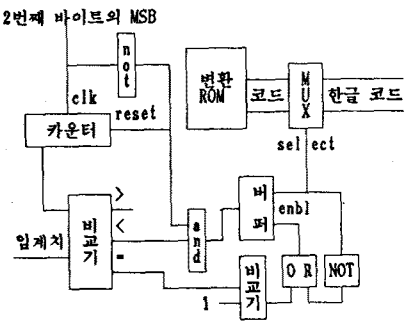


그림 4. 조합형 코드 변환 논리

해 보았다. 본 한글 디코더의 장점은 사용자가 어떠한 코드 체계를 쓰던지간에 본 한글 디코더로 코드 체계를 스스로 인식하여 적절한 완성형 한글로 표시하였다는 것이다. 따라서, 기존에 있는 한글 소프트웨어가 본고에서 제안한 한글 디코더에서는 어려움없이 수행될 수 있다는 것이다. 본 고에서 고려하지는 않았지만 앞으로의 과제는 현재의 한글을 표시할 때 그 코드는 한글 디바이스 드라이버를 의운다던지 RAM 상주 프로그램에 의해서 생성되어 원천적으로 한글 코드임이 밝혀져 있지만 한글 코드를 생성하는 부분도 한글 카드에서 처리해준다면 영숫자를 표시할 때와 한글을 표시할 때의 표시 성능차를 어느 정도 극복할 수 있도록 하는 것도 중요하리라 본다.

참고 문헌

1. 한국표준연구소, 한글/한자 정보 처리기술 핸드북, 과학기술처, 1988.

2. 이석호, 송병호, 컴퓨터 처리를 위한 한글 코드, 서울대학교, Feb., 1987. pp 87-93.

3. 이병태, 송영해, CRT 디스플레이 기법, 기전연구사, 1983.

4. 테크니컬 그룹, "IBM PC 비디오상에서 한글구현", 컴퓨터 매거진, July, 1988, pp 168-170.

5. 테크니컬 그룹, "MDA, HGC, CGA 한글 설계", 컴퓨터 매거진, August, 1988, pp 164-170.

6. Krutz, Interfacing techniques in Degital design, jhon wiley & sons, 1988,pp 41-83.

7. IBM, Technical Reference Personal Computer XT, 1983, pp 113-150.

8. IBM, Technical Reference Personal Computer AT, 1986, pp 25-37.