論文 82-19-5-6

어드레스 變換 技法과 DMA를 이용한 英文/한글/漢字 디스플레이 콘트롤러 設計

(English/Hanguel/Chinese Character Display Controller Design Using Address Conversion Technique and DMA)

金 昌 滿*, 黃 熙 隆**

(Chang Man Kim and Hee Yeung Hwang)

要 約

라스터 스캔닝 CRT 그라픽 방법을 사용하여 英文/한글/漢字 디스플레이 콘트롤러 를 設計하는 경우에, DMA(direct memory access)와 어드레스 변환 기법을 이용하여 설계를 간단히 할 수 있었으며 실제로 64자×16줄 화면 구성의 설계 예를 보였다.

Abstract

This paper shows a design method of English/Hanguel/Chinese display controller using address conversion thchnique and DMA in the raster scanning graphic CRT display by giving a design example (64 characters × 16 lines display controller).

I. 序 論

라스터 스캔닝 그라픽 CRT 디스플레이를 사용하여 英文/한글/漢字 컴퓨터 디스플레이 터미날 또는 워드-프로세서 등에 응용을 하고저 하는 경우, 디스플레이 속도와 그 제어의 용이성이 문제가 된다. 특히 高速으로 데이타 송수신을 행하는 한글/漢字 컴퓨터 디스플레이 터미날인 경우에는 디스플레이 속도를 충분히 고려하여야 한다. 그라픽 디스플레이 방법을 사용하여 漢字를 디스플레이 시키기 위해서는 먼저 Font (字型) ROM (read only memory)에 저장되어 있는 Font의 데이타를 읽어서 이것을 디스플레이 리후레쉬 메모리의 해당 위치에 라이트 해야 한다. 이러한 절차가 디스플레이 제어 소프트웨어에 의해서만 행해지면 디스플레이 속도가 느려질 뿐만 아니라, 그 제어도 매우 복잡해진다. 이러한 문제점을 해결하기 위해 그림 1과 같이 既存의 그라픽 콘트롤러에 어드레스 變換 技法과 DMA를 이용하여 디스플레이 속도를 증가시킬 수 있었으며 또한 디스플레이 제어도 용이하게 할 수 있었다.

한글과 漢字는 日本語와는 달리 음절이 항상 1:1로 대응한다. 이 固有한 特性을 사용하면 효율적, 경제적인 한글/漢字 入力/出力 시스템을 설계할 수 있다(이러한 방법을 同音 異義入力 방법이라 한다.).

本 논문에서는 同音 異義 入力 방법을 사용하여 64字×16줄의 英文/한글/漢字 디스플레이 콘트롤러를 설계하는 경우, DMA와 하드웨어 어드레스 변환을 導入하여 디스플레이 속도 증가와 그 제어를 용이하게

*, **正會員, 서울大學校 工科大學 電子計算學科
(Dept. of Computer Eng., Seoul National Univ.)

接受日字 : 1982年 5月 17日

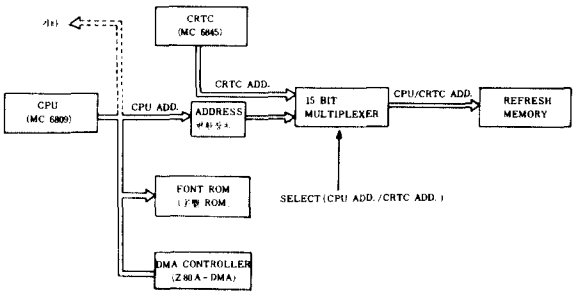


그림1. DMA와 어드레스 變換을 위한 버지 연결도

Fig. 1. Address bus block diagram using DMA & address conversion.

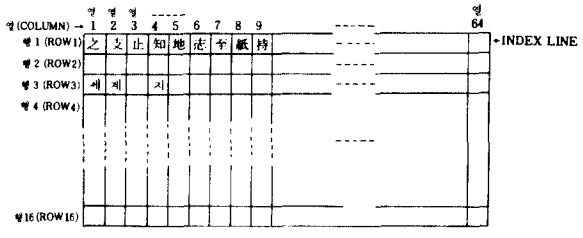


그림 2 CRT 화면상의 인덱쓰 라인에 디스플레이 되는 同音 異義 漢字(예 : "지")

Fig. 2. Phonetically same Chinese character display on index line of CRT display (example : "gi").

하는 방법에 대하여 연구하였고, 이 방법으로 設計 製作하여 結果를 確認하였다.

Ⅱ. CRT 화면上의 디스플레이 速度比較
(既存 소프트웨어에 의한 方法과 DMA방법)

同音 異義 入力 方法에 의한 한글/漢字 디스플레이 콘트롤러를 설계하는 경우, 文字의 디스플레이를 既存의 방법대로 소프트웨어로 처리할 때와, DMA를 사용할 때의 디스플레이 시간을 비교해 본다.

1. 同音 異義 入力 方法과 고려한 점

한글 타이프라이터 키보드를 사용하여 漢字 정보를 입력하려면, 먼저 한글 자모키를 쳐서 입력하고져 하는 漢字와 대응되는 한글을 모아쓰기 한 다음, 그에 해당되는 모든 同音 異義 漢字를 인덱쓰 라인에 나타나게 하여, 그 중에서 하나를 선택하게 된다. 예를 들어 漢字入力 모드에서 한글 "지"를 치면 그림2와 같이 행 1 (인덱쓰 라인으로 사용)에 "지"의 同音 異義 漢字가 Font ROM에 저장된 순서대로 모두 나타난다. 예를 들어서 "地"를 入力코져 하면 번호 "5"를 치게 된다. 그러면 디스플레이 되었던 한글 "지"는 "地"로 바뀌게 된다.

이와 같은 방법으로 漢字 入力을 하는 경우 인덱쓰 라인에 디스플레이 되는 同音 異義 漢字들의 디스플레이 속도가 최대 키보드 입력 속도보다 느린 경우 디스플레이가 끝날 때까지 키보드 입력을 늦추어야 될 경우가 있다.

최대 키-보드 입력 속도는 10ms 정도이기 때문에 인덱쓰 라인 디스플레이가 최대한 10ms 이내에 이루어지도록 설계 되어야만 한다.

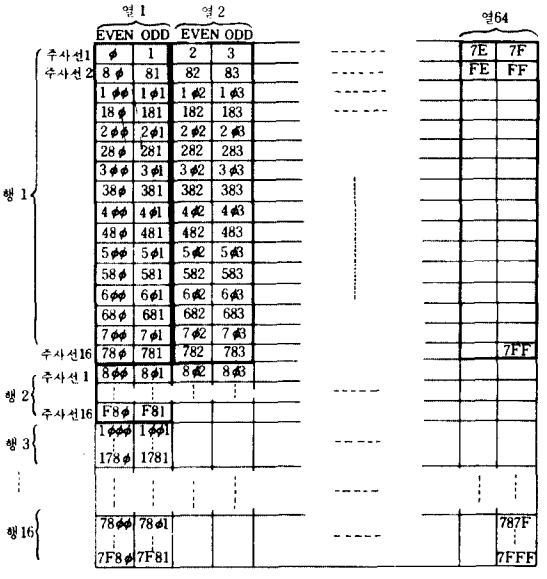


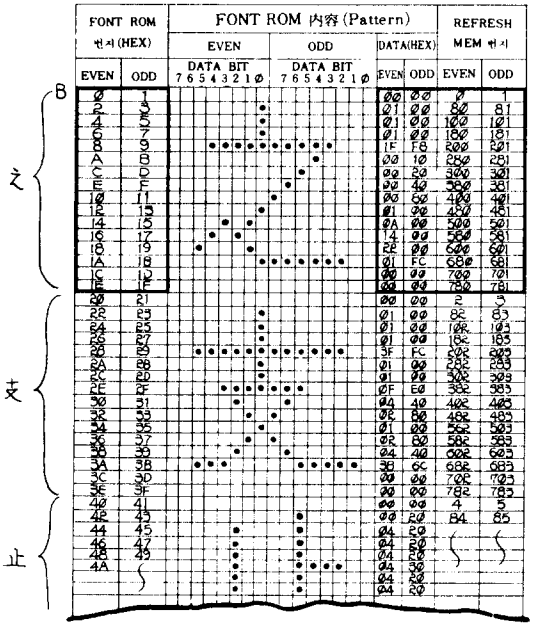
그림3. 既存 라스터 스캔닝 그라픽의 CRT 화면 위치와 리후레쉬 메모리 어드레스 관계도

Fig. 3. Relation between CRT display position & refresh memory address in general purpose raster scanning graphic.

일반적으로 라스터 스캔닝 그라픽 디스플레이 방법에서 화면의 위치와 리후레쉬 메모리의 관계는 그림 3과 같이 리후레쉬 메모리 어드레스가 수평방향으로 증가하다가 끝나면 1주사선 증가한 후 다시 수평으로 증가하는 과정을 되풀이 하면서 리후레쉬 메모리에에 저장된 비트 패턴을 읽어 내어서 디스플레이 시킨다.

그런데 Font ROM에 저장되어 있는 Font의 저장 방법은 그림 4 에서와 같이 漢字의 글자 단위로 모든 同音 異義 漢字들이 연속하여 저장되어 있다. 그림 4

에서 보면 Font의 데이타를 나타내는 Font ROM 어드레싱과 CRTC(CRT 콘트롤러)어드레싱이 서로 다르다. DMA를 사용하지 않고 Font를 디스플레이 시키기 위해서는 Font 데이타를 차례로 읽어 내어 해당되는 리후레쉬 메모리 어드레스로 소프트웨어에 의해서 변환한 다음 라이트 해야 한다.



CRTC ADD선택

그림4. Font ROM과 리후레쉬 메모리 관계(B는 "지" 同音 異義 漢字가 시작하는 Font ROM 시작 번지)

Fig. 4. Relation between Font ROM and refresh memory (B denotes the starting address of Font "gi" in Font ROM).

2. 소프트웨어에 의한 디스플레이 시간

소프트웨어에 의해서 디스플레이 시키는 경우에 그 속도를 계산해 보기 위해 한 文字를 Font ROM에서 리후레쉬 메모리에 옮기는 어셈블러 프로그램을 작성하면 다음과 같다. ( )안의 수는 머쉰 싸이클 수를 나타낸다.

```
        LDA   #16        A REG=16(2)
        STA   COUNT      COUNT=16(4)
LOOP    LDD   ,X++       X REG=ROM FONT ADD, X←X+2 AFTER EXECUTION(6)   } 2 BYTE MOVE
        STD   0,Y        Y REG=REFRESH MEMORY POINTER (5)               }
        LEAY  80,Y       Y=Y+80 : NEXT REFRESH MEMORY ADJUST (5)
        DEC   COUNT      DECREMENT COUNTER (6)
        BNE   LOOP       COUNT=0 ? (3)
        RTS              1 字 DISPLAY 完료(5)
```

아래 어셈블러 프로그램은 MC6809의 mnemonic 코드이며, 1MHz 클럭을 사용한 경우 머쉰 싸이클 타임은 1마이크로-쎄컨드(μsec)가 된다. 레지스터 X, Y는 각각 Font ROM 어드레스와 리후레쉬 메모리 어드레스를 가리키는 어드레스 포인터 레지스터이며 이미 그 값들이 설정되어 있다고 가정하면 1 漢字 Font를 디스플레이 시키는 시간 T_{P1}은

$T_{P1}=2+4+16\times(8+5+5+6+3)+5=443$ (μs)이며 30 Font를 인덱쓰 라인에 디스플레이시키는 시간 T_{P30}

$T_{P30}=443\times30=13,290$ (μs)이 되며 60Font인 경우는 $T_{P60}=443\times60=26,580$ (μs) ≒27ms 가 된다.

그런데 인덱쓰 라인에 디스플레이 되는 同音 異義 漢字수는 종류에 따라서 최대 60字까지 되므로 최악의 경우 디스플레이 속도가 최대 키-보드 입력 속도보다 느려지게 된다.

3. DMA를 사용한 경우 디스플레이 시간

이러한 문제점을 해결하기 위해서 DMA 방식을 사용하여 memory to memory date move를 행해야 한다.

1마이크로 쎄컨드에 1바이트를 옮기는 DMA 콘트롤러 (Z80A-DMA)를 사용하는 경우 30Font를 옮겨서 디스플레이 하는 데 소요되는 시간

$T_{D30}=32\times30=960$ (μs)이며, 60Font인 경우

$T_{D60}=32\times60=1,920$ (μs)가 된다.

이와 같이 DMA를 사용하면 최악의 경우에도 최대 키-보드 입력속도(10ms)보다 적게 되므로 디스플레이 속도 문제는 완전히 해결된다.

그런데 DMA로 메모리에서 메모리로 데이타를 옮기는 경우 DMA어드레스가 리니어하게 변하기 때문에, 어드레스 변환을 행하지 않고 DMA memory to memory move를 행하게 되면 Font 데이타는 글자 모양대로 디스플레이 되지 않고 주사선을 따라 수평으로 흩어져서 디스플레이 되므로 아무런 의미를 갖지 못하게 된다.

이 문제를 해결하기 위해서 어드레스 變換을 해야 한다.

Ⅲ. DMA와 어드레스 변환에 의한 한글/漢字 디스플레이

디스플레이의 위치를 리후레쉬 메모리와 1 : 1로 대응하는 리후레쉬 메모리의 절대 어드레스라 할 때, 그 절대 어드레스의 데이타를 액세스하는 CPU 어드레스와 CRTC 어드레스는 각각의 어드레스 變換을 거쳐 액세스 할 수 있다. 그러나 CRTC 어드레스와 디

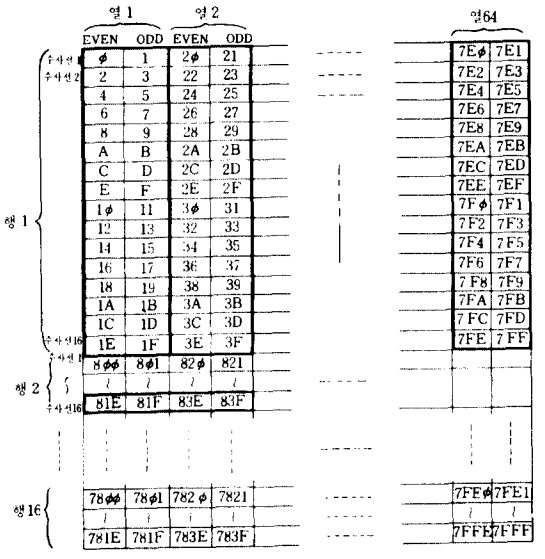


그림5. 어드레스 變換후의 디스플레이 위치와 CPU 어드레스 관계표(Hexadecimal 표시)

Fig. 5. Relation between display position and CPU address after address conversion (Hexadecimal notation).

스플레이 위치가 그림 3과 같이 1 : 1로 대응되는 경우 CRTC 어드레스는 변환을 하지 않아도 된다.

CPU어드레스(=DMA 어드레스)와 디스플레이 위치의 관계가 그림 5와 같다고 하면 이런 형식은 Font ROM에 저장되어 있는 Font의 형식과 같으므로 DMA를 사용하여 同音 異義 漢字 블럭 메모리를 그대로 리후레쉬 메모리에 차례로 옮겨 놓기만 하면 된다. (한글 디스플레이는 한글 모드에서 입력된다. 한글 Font ROM에는 한글 자모 (예 : ㄱ ㄴ ㅏ ㅑ 등)가 각각 漢字 1字와 같은 크기로 저장되어 있다. 한글 모드에서의 디스플레이는 키-보드로 부터 입력되는 자모의 해당 Font 데이타들이 소프트웨어에 의해서 로지컬 OR로 모아쓰기 조합된 후, 이 결과를 DMA에 의해서 리후레쉬 메모리의 해당 위치에 옮기게 되며, 영문 모드에서 영문 기호 디스플레이는 漢字와 같이 Font ROM에 있는 Font 데이타를 DMA에 의해 해당 리후레쉬 메모리에 옮기게 된다.)

이렇게 하기 위해서는 CPU 어드레스가 멀티플렉서에 의해서 선택될 때 그 어드레스가 변환되어야 한다.

예를 들어서 디스플레이 행 1/열 1/주사선 16/Even 위치에 해당되는 리후레쉬 메모리를 악세스하기 위해서 CRTC는 어드레스 780(Hexadecimal)으로 악세스하고 DMA 어드레스는 1E(Hexadecimal)로써 악세스하게 된다.

1. 어드레스 變換 방법

이와 같이 서로 다른 어드레스로 同一한 리후레쉬 메모리를 악세스 하기 위해서는 CPU 어드레스가 멀티플렉서에 도달하기 전에, 입력 어드레스 라인과 같은 수의 어드레스 라인이 출력되는 하드웨어 변환을 사용하면 된다. 변환 방법으로써 룩업 테이블 ROM을 사용하는 경우 64字×16줄/화면

16 dot × 16 dot / 文字

로 구성된 디스플레이 리후레쉬 메모리의 크기는 16×2×64×16=32,768(byte)가 되며 이에 해당되는 어드레스는 $A_0 \sim A_{14}$가 된다. 따라서 변환 ROM은 그림 6과 같이 구성 된다.

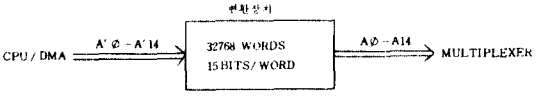


그림6. 변환 방법으로써 ROM을 사용한 경우

Fig. 6. Address conversion with ROM.

그러나 이러한 방법은 변환 ROM의 크기와 비용이 너무 커져서 실용성이 없어진다. 이것을 개선하기 위해서 디스플레이 구성을 다음과 같이

열수/화면 = 2^n (단, n은 정수)

주사선수/행 = 2^m (단, m은 정수)

로 하면 어드레스 변환은 그림 7과 같이 어드레스 라인의 연결을 교차시킴으로써 간단히 이루어 질 수 있다. 그림 7은 64열/화면, 16 주사선/행 인 경우에 연결도를 나타낸다.

그림 7과 같이 입력과 출력 어드레스를 배열하면 CPU 어드레스와 디스플레이 위치 관계는 그림 5와 같이 되며 CRTC 어드레스와 디스플레이 위치는 그림 3과 같이 된다.

그림 8은 구체적인 어드레스 변환과 그 연결도를 나타낸다.

2. 어드레스 變換 예 (그림 8 참조)

CPU 어드레스가 멀티플렉서에 의해서 선택 되었을 때 그 어드레스가 1F(Hex)였다면 멀티플렉서를 거친 후의 CPU 어드레스는 (가)와 같다.

멀티플렉서를 거친 후에, (가)의 어드레스와 대응되는 CRTC 어드레스 MA'Ø~MA'14는 (나)와 같다.

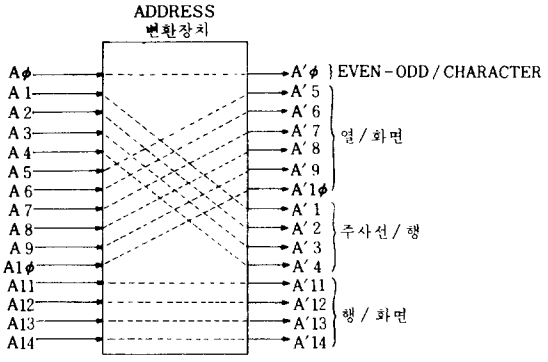


그림7. 64열/화면, 16주사선/행인 경우 변환 장치 (어드레스 라인 교차에 의한 번지 변환 방법)

Fig.7. Address conversion in 64column/display, 16 scan line/row organization (only the address line connections are changed).

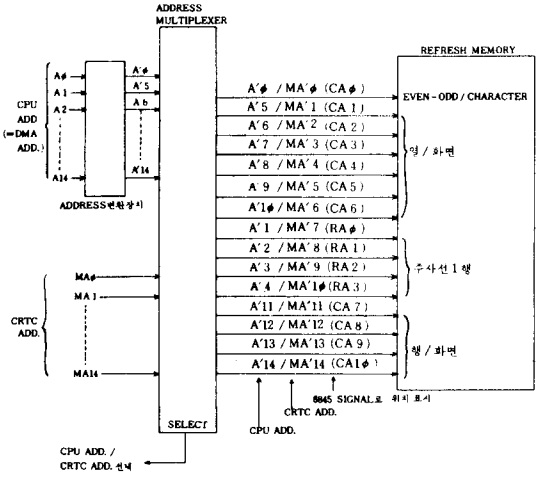


그림8. 구체적인 어드레스 연결도

Fig.8. Detailed address connection.

그런데 CRTC 어드레스는 어드레스 변환을 행하지 않았기 때문에 MA∅~MA14와 MA'∅~MA'14는 순서가 같다. 이를 다시 CRTC 어드레스 순서로 배열하면 (다)에서 781(HEX)이 된다. (그림3과 그림5 비교) 이와 같이 CPU 어드레스와 CRTC 어드레스는 리후레쉬 메모리를 통하여 1 : 1로 대응하게 된다.

이런 방법의 어드레스 변환에는 또 다른 장점이 있다. 그림5에서 살펴보면 디스플레이 되는 각 문자 위치의 첫번째 어드레스는 일정한 크기(Font 데이타 만큼)로 먼저 열(column) 방향으로 차례로 증가한 후 끝나면 다음 행(row)에서 다시 열방향으로 증가하게 된다.

Font의 데이타가 32바이트로 구성되어 있는 경우 CPU 어드레스는 각각 다음과 같은 블럭으로 나눠진다.

A_0~A_4 ; Font 데이타

A_5~A_{10} ; 열(column) 위치 표시

A_{11}~A_{14} ; 행(row) 위치 표시

따라서 디스플레이 上의 어떤 위치에 문자를 디스플레이 시키고져 할때는 열 어드레스(A_5~A_{10})와 행 어드레스(A_{11}~A_{14})를 설정한 후 해당 Font 데이타를 A_0~A_4에 차례로 옮겨 놓기만 하면 된다.

이러한 방법으로 문자의 위치를 지정할 수 있게 되면, 화면상에서 문자의 삽입, 제거 등을 제어하는 디스플레이 제어 프로그램도 간단해진다.

Ⅳ. 結 論

라스터 스캔닝 CRT 그라픽 디스플레이에 DMA와 어드레스 變換기법을 도입하여 英文/한글/漢字 디스플레이 콘트롤러를 設計, 製作한 결과 다음과 같은 結論을 얻었다.

1) 既存의 CRT 그라픽 콘트롤러를 사용하여 소프트웨어에 의해서 한글/漢字를 디스플레이 시키는

(가)

CPU(DMA) ADD	A'14	A'13	A'12	A'11	A'10	A'9	A'8	A'7	A'6	A'5	A'4	A'3	A'2	A'1	A'0
HEX(1F)	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	1	1	1	1	1

(나)

CRTC ADDR	MA'14	MA'13	MA'12	MA'11	MA'6	MA'5	MA'4	MA'3	MA'2	MA'1	MA'10	MA'9	MA'8	MA'7	MA'0
ADDR PATTERN	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	1	1	1	1	1

(다)

CRTC ADD	MA 14	MA 13	MA 12	MA 11	MA 10	MA 9	MA 8	MA 7	MA 6	MA 5	MA 4	MA 3	MA 2	MA 1	MA 0
ADD. PATTERN (HEX781)	∅	∅	∅	∅	1	1	1	1	∅	∅	∅	∅	∅	∅	1

경우에 발생하는 시간 지연 문제를 DMA와 어드레스 변환 기법을 도입하여 완전히 해결하였음.

2) 어드레스 변환 기법을 사용함으로써 화면 전체를 1文字 크기 블럭으로 分割 可能하다. 또 文字 지정 어드레스가 列, 行 順으로 증가하기 때문에 디스플레이 제어 프로그램이 월등하게 간단해진다.

3. 英文도 漢字와 같은 방법으로 디스플레이 시키므로 英文/한글/漢字 혼용이 가능하다.(英文, 한글, 漢字 모두 같은 크기 32 바이트로 디스플레이 함.)

4) 文字 블럭의 크기를 프로그램어블 하게 어드레스 변환 부분의 입력/출력 조합을 소프트웨어로 제어하도록 하면 화면의 구성을 자유로 바꿀 수 있어 다양한 그라픽 응용이 가능하다.

參 考 文 獻

[1] William M. Newman, Robert F. Sproull, *Principles of Interactive Camputer Graphics*. McGraw-Hill, New York pp. 211-287, 1979.

[2] Mc6809 Data Sheet, Motorola, 1980.

[3] Mc6809 Programming Manual, Motorola, 1979.

[4] Z80A-DMA Data Sheet, Zilog, 1980.

[5] Rodnay Zaks, Austin Lesea, *Microprocessor Interfacing Technique*. Sybex, 1979.