

한글 외곽선 글자체 생성 가속기의 설계 및 구현

(Design and Implementation of Hanguk Outline Font Generation Accelerator)

裴 鍾 洪*, 黃 奎 哲**, 李 潤 泰**, 慶 宗 旻**

(Jong Hong Bae, Gyu Cheol Hwang, Yun Tae Lee, and Chong Min Kyung)

要 約

본 논문에서는 한글 포스트스크립트와 같은 외곽선 글자체 문자에 대한 정보로부터 점행렬 데이터를 생성하여 화면과 LBP에 출력하는 하드웨어 가속기를 설계하고 이를 구현하였다. 전체 시스템은 연산단과 출력단으로 구성되어지며 AT보드 2장 크기의 PC/AT 응용 보드로 구현되었다. 연산단은 마스터 프로세서(MC 68000)와 MC 56001 그리고 곡선 계산 전용 ASIC 칩인 KAFOG를 이용하여 2개의 슬레이브 프로세서로 구성된다. 출력단에서는 화면의 제어를 위해서 TMS34061을 사용하였고, LBP 엔진의 제어를 위해서는 GP425를 사용하였다. 사용된 모니터의 해상도는 640 × 480이고, LBP엔진의 해상도는 2385 × 3390이다. 지금 KHGB90-B로 명명된 이 시스템은 32 × 32 문자를 초당 약 100자 생성한다.

Abstract

In this paper, we designed and implemented a hardware accelerator for the generation of bit map font from Hanguk outline font description for LBP (Laser Beam Printer) and screen applications. Whole system was implemented as a double size PC/AT application board which consists of processing block and display block. The processing block has a master processor (MC68000) and two slave processors which are MC56001 and KAFOG chip responsible for the short vector generation. In the display block, TMS34061 was used for monitor display and GP425 was used for LBP print out. The resolution of the monitor is 640 X 480 and that of LBP is 2385 X 3390. The current system called KHGB90-B generates about 100 characters per second where each character consists of 32 X 32 bits

I. 서 론

컴퓨터와 반도체 기술이 발전함에 따라 데이터의

처리속도가 빨라지고 기억용량이 증가하게 되었으며, 이러한 발전은 현대사회에서 엄청나게 늘어나는 정보를 정확하고 빠르게 처리하는 것을 가능하게 했다. 최근에는 컴퓨터가 데이터나 정보를 수치적으로 처리하는 단계에서 벗어나 그래픽 기능이 강화되는 등 그 기능이 무척 다양해져서 사용범위가 급격히 늘어나게 되었으며 이에 힘입어 LBP(laser beam printer), 워드프로세서등을 이용하는 전자출판 시스템이 등장하게 되었다.

이와 같은 전자출판 시스템에서 이용되는 글자체

*準會員, **正會員, 韓國科學技術院 電氣 및 電子工學科
(Dept. of Electrical Eng., KAIST)

接受日字: 1991年 7月 29日

(※ 본 논문은 상공부의 지원으로 (주)휴먼 컴퓨터의 협력하에 수행되었던 공업기반기술 개발사업의 연구결과의 일부분임을 밝힙니다.)

에는 점행렬 글자체(bitmap font), 외곽선 글자체(contour font), 구조적 글자체(structural font) 등^{3,4)}이 있고, 이들 글자체의 선택은 출력 장치의 특성, 응용 소프트웨어 및 사용자의 요구조건에 따라 결정된다.

점행렬 글자체는 텍스트용 화면이나 점행렬 프린터에 많이 사용되는 글자체로써 이를 이용하면 출력 속도는 빠르지만, 고해상도 출력장치이거나 한글 및 한자에서와 같이 사용되는 글자의 집합이 클 경우에는 많은 기억장소를 필요로 하므로 비효율적이다. 또한 이 방법으로는 다양한 변화를 주어서 글자를 생성하는 데는 많은 어려움이 있다.

외곽선 글자체에서는 글자체 외곽선을 점들의 집합, 원호, 또는 자유곡선등으로 표현한다. 이 방식으로 설계된 글자는 글자를 생성할 때, 위의 방법보다 많은 시간이 소요되지만, 고해상도 출력 장치에 보다 정교한 글자를 나타내고, 글자의 크기 변화나 인쇄 방향등의 변화에 따라 상응하는 글자들을 정확하게 생성할 뿐만 아니라 글자의 설계도 훨씬 수월하여 PostScript, Impress등에서 많이 사용된다.

현재 전자 출판 시스템에 사용되는 외곽선 글자체로 된 영문자나 숫자, 심볼들은 갯수의 글자가 반복적으로 사용되고, 글자 구조도 간단하여 지금까지 개발된 범용 프로세서를 이용하여 소프트웨어로도 처리가 가능하다. 하지만, 한글이나 한자는 글자체의 구조가 복잡하고 사용되는 글자의 수도 많아 이 글자체를 전자 출판에서 사용할 경우에는 고속의 문자 처리용 그래픽 시스템이 요구된다.

본 논문은 외곽선 글자체로 한글을 생성할 경우, 생성 속도에 의해서 발생하는 문제점들을 해결하기 위하여 한글 생성을 가속화 하는 병렬 처리 하드웨어 구조의 제안과, 설계에 관한 것이다.

II. 시스템 개요

KHGB90-B 시스템은 호스트를 PC/AT로 하여 PC/AT의 add-on 보드로 사용되어지는 시스템으로서, KHGB90-A²⁾의 여러가지 문제점들을 개선한 시스템이다. KHGB90-A에서는 발생한 문자들의출력이 모니터에 국한되었다. 그리고, 이 시스템은 호스트로 이용되는 PC/AT를 확장 메모리 영역(extended memory range)으로 사용할 때의 전체 메모리 크기인 16 Mbyte를 모두 다 사용하기 때문에 다른 시스템을 PC/AT에 add-on형식으로 같이 사용할 수 없는 단점을 가지고 있었다. 또한, 이 시스템의 문자발생 속도는 약 4 CPS(characters per second)인데, 이러한 속도는 8ppm(pages per minute)의 LBP 엔

진을 기준으로 할 때 엔진을 최대로 지원하기 위하여 요구되는 문자 발생 속도인 250 CPS와는 상당한 차이가 있다.

KHGB-90B는 발생한 문자를 모니터에 일차적으로 출력한 후, 사용자의 요구가 있을 경우에는 LBP(laser beam printer)에 출력하는 방식을 사용하는 시스템이다. 그리고, 상기의 메모리 맵에서의 문제들을 해결하기 위하여 KHGB90-B에서는 세그먼트레지스터 방식을 이용하여 PC/AT의 전체 메모리 영역인 16Mbyte중에서 32Kbyte만으로 이 시스템과의 인터페이스를 가능하게 설계하였다. 또한, 문자 발생 속도에서 제한요인이 되었던 슬레이브단을 강화하기 위하여 슬레이브 1프로세서로는 DSP(digital signal processing) 칩인 MC56001을 사용하였다.

1. 전체 시스템의 하드웨어 및 소프트웨어 구조
KHGB90-B시스템의 전체적인 하드웨어 및 소프트웨어의 구조는 그림 1에 나타난 것과 같다. 먼저, 사용자들에 의해서 문서편집과 같은 일들은 호스트인 PC/AT에서 수행된다. 이 편집된 자료들은 PostScript 형태의 ASCII 파일로 저장되고, PC/AT에서 8비트의 통신용 레지스터를 통해서 이 시스템의 상태를 점검한 후, 데이터를 보낼 수 있다고 판단이 되지만, 파일 단위로 이 시스템의 마스터 메모리 데이터 영역으로 데이터가 전송된다. 데이터 전송이 끝나면 마스터 프로세서는 보내어진 데이터를 해석한다. 데이터를 해석할 때 폰트 생성이 요구될 때에는 슬레이브단에게 이를 부탁하고, 폰트 생성이외의 그래픽 기능은 마스터 프로세서가 직접 행하게 된다. 폰트의 비트 맵(bit map) 데이터가 슬레이브단에 의해서 모두 만들어지면 마스터 프로세서가 이 데이터를 모니터로 출력하고, 사용자의 요구가 있을 경우에는 LBP 엔진에 출력한다. 이 때, LBP에 출력되는 데이터는 모니터에 출력되는 데이터와 해상도가 다르기 때문에 다시 생성되어야 한다.

2. 마스터 프로세서의 처리과정

마스터 프로세서의 처리과정은 그림 2의 흐름도에 나타난 것과 같다. 폰트 생성이 요구되면, 마스터 프로세서는 먼저 슬레이브단을 액세스(access)할 수 있는지를 점검하는데, M/S 플래그(마스터와 슬레이브 사이의 플래그)가 1이면 슬레이브 1 프로세서를 홀트(halt)시킨 후, 폰트를 나타내는 직선과 곡선의 정보를 담고 있는 점들(control point)로 구성된 폰트 데이터를 폰트 메모리에서 가지고와서 슬레이브 1의 데이터 메모리 영역으로 옮긴다.

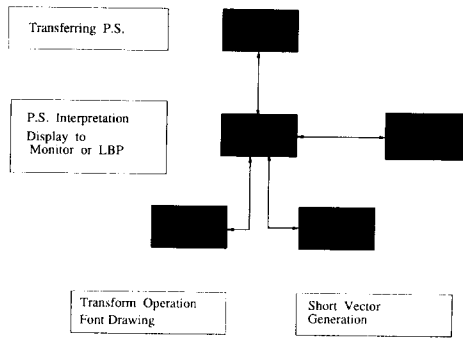


그림 1. 전체 시스템의 하드웨어 및 소프트웨어의 구조

Fig. 1. Hardware and software architecture of whole system.

일들을 수행한다. 매스터 프로세서는 슬레이브 2프로세서에게 외곽선을 그리기 전에 폰트 외곽선의 중간값의 계산을 요구한다. 이러한 과정을 중간값(short vector) 생성과정이라고 하며, 매스터 프로세서는 슬레이브 1프로세서에 의해서 변형된 폰트 데이터를 슬레이브 2프로세서에게 줌으로 중간값을 구한다. 이후, 매스터 프로세서는 status를 DRAW로 한 후에 슬레이브 1프로세서가 폰트 생성을 위한 나머지 작업을 할 수 있게 슬레이브 1프로세서의 홀트를 풀어준다.

이 과정 후에, M/S 플래그를 확인하여 1이면 슬레이브 프로세서를 홀트시킨후, 슬레이브 프로세서에 의해서 생성된 폰트의 점행렬 데이터를 Frame buffer로 옮긴 후, PS 데이터의 해석을 하고 앞에서와 같은 일련의 과정을 반복한다.

3. 슬레이브 1 프로세서의 처리과정

슬레이브 1 프로세서가 하는 일은 크게 TRANS과정과 DRAW과정 2가지가 된다. TRANS과정은 scaling factor와 좌표변환 각도를 이용하여, 폰트를 변환하는 과정이고, DRAW과정은 매스터와 슬레이브 2 프로세서에 의해서 생성된 중간값을 이용하여 폰트의 외곽선을 생성하고 필요에 따라서는 filling이 된 폰트 데이터를 생성하는 과정이며, 그 과정은 그림 3에 나타난 것과 같다.

먼저, 슬레이브 1 프로세서는 M/S 플래그가 0이 되기를 기다린다. M/S 플래그가 0이 되면 status를 통해서 TRANS과정과 DRAW과정을 구분해주고, 각각에 대한 처리를 한다. 이 과정 후에 M/S 플래그를 1로 하여서, 매스터 프로세서가 다음 과정을 할 수 있게 한 후, M/S 플러그가 0이 되기를 기다린다.

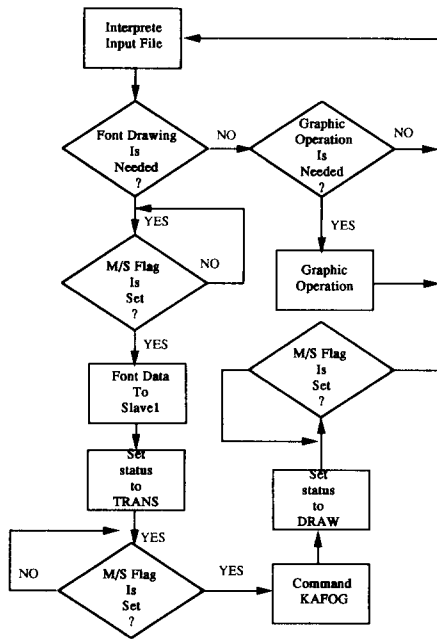


그림 2. 매스터 프로세서의 처리과정
Fig. 2. Flow chart of master processor.

이 과정을 마친 후, 매스터와 슬레이브간의 통신을 위해 할당된 슬레이브 메모리내의 status를 TRANS로 한 후에 슬레이브단에 홀트를 해제하여 슬레이브단이 scaling과 좌표변환의 과정을 하게 한다.

이 후, 매스터 프로세서는 M/S 플래그가 1이 되기를 기다리고, 1이 된 것이 확인되면 슬레이브 1 프로세서를 홀트시킨 후에 슬레이브 2프로세서와 다음의

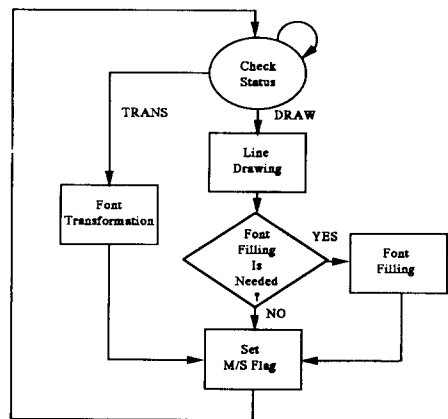


그림 3. 슬레이브 1 프로세서의 처리과정
Fig. 3. Flow chart of slave1 processor.

Ⅲ. 전체 하드웨어의 구조

전체 하드웨어 시스템의 블록 다이어그램은 그림 4에 나타난 것과 같이 호스트 인터페이스부, 연산단 그리고 출력단으로 나누어진다.

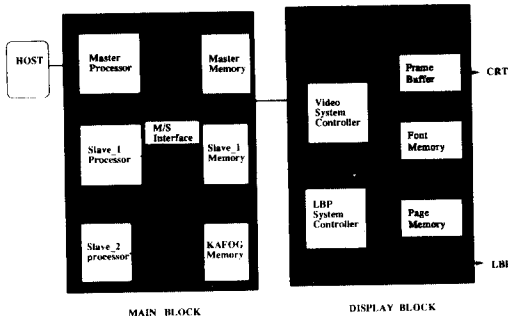


그림 4. 전체 하드웨어의 블록 다이어그램
Fig. 4. Block diagram of hardware system.

1. 호스트 인터페이스

호스트 인터페이스에 있어서 KHGB-90-A는 매스터와 슬레이브의 프로그램다운 로딩(down loading)과 폰트 메모리 다운 로딩등의 일들에 국한시켰다. 이 일들을 수행하기 위해서 KHGB90-A는 확장 메모리 영역을 이용하여 PC/AT가 사용할 수 있는 16 Mbyte를 거의 다 사용하였다. 따라서, 다른 시스템을 PC/AT에 같이 사용할 때 메모리 맵에 상당한 문제가 있었다.

KHGB-90B에서는 하드웨어 디버깅을 용이하게 하도록 시스템의 모든 디바이스들을 호스트에서 액세스할 수 있게 설계하였다. 또한, 이러한 일들을 하기 위해서는 16Mbyte의 메모리 영역이 필요한데 세그먼트 레지스터를 이용하여서 실제적으로 사용한 메모리 크기는 32Kbyte이다. 이 시스템에서 사용한 메모리 구조는 그림 5에 나타난 것과 같다.

세그먼트 레지스터를 이용하여서 호스터 인터페이스하는 과정은 다음과 같다. PC/AT의 메모리 영역중에서 일반 사용자들이 사용할 수 있는 영역은 0X0c800~0x0dfff이다. 이 메모리 영역중에서 0X0dd000~0x0d7ff의 메모리 영역에 해당하는 하위 15bit를 레지스터의 오프셋(offset)으로 이용하고, 상위 어드레스에 해당하는 9-bit 레지스터는 0X0d800번지에 의해서 정해진다. 이 때, 호스트와 이 시스템과의 통신은 모두 워드(word)단위로 행해진다. 이를 위해서 데이터 전송이 요구될 때, PC/AT의 MEM-CS16 신호를 LOW로 구동시켜 이를 가능하게 하였다.

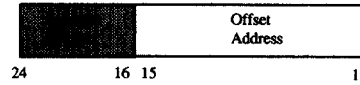


그림 5. 호스트의 메모리 구조
Fig. 5. Memory structure of Host.

2. 연산단

연산단에는 앞 장에서 설명했던 폰트 생성에서 필요한 주요 과정들을 수행하는 단으로서, 크게 매스터단, 슬레이브 1단 그리고 슬레이브 2단으로 나누어진다.

(1) 매스터단

매스터단은 매스터 프로세서와 매스터 메모리로 구성되어진다. 매스터 프로세서는 DRAM(dynamic random Access Memory)을 포함해서 여러가지의 디바이스(device)들을 액세스해야 하므로 비동기 액세스(asynchronous access)가 가능하며, 8Mbyte 이상의 메모리 영역을 가져야 하는데 MC68000은 16Mbyte의 충분한 메모리 영역을 갖고 있으며, 비동기 액세스가 가능한 프로세서이므로 매스터 프로세서로 사용된다. 그리고 매스터 메모리는 포스터 스크립트 해석(Post-Script intpretation) 프로그램 및 PC/AT에서 온 포스터 스크립트 데이터를 저장하기 위해서, 128Kbyte의 메모리 공간이 필요한데, 이것은 비교적 작은 메모리 영역이므로, 액세스가 간편한 SRAM(Static Random Access memory)으로 구현하였다.

(2) 슬레이브 1단

II 장에서 설명한 슬레이브 1 프로세서의 일들을 잘 해내기 위해서는 곱셈기능과 메모리 액세스를 빨리 행할 수 있는 프로세서가 필요한데, 내부에 곱셈기를 가지고 있으면서, 멀티 데이터 버스(multi data bus)를 통해서 데이터를 페치(fetch)함으로써 병렬도를 높일 수 있는 DSP 칩, MC56001을 슬레이브 1 프로세서로 사용하였다.

슬레이브 1 메모리는 슬레이브 1이 20MHz에서 동작할 때, 이것을 충분히 지원해 주기 위해서는 35nsec 이하의 액세스 타임(access time)을 갖는 SRAM으로 구현해야 되는데, 메모리 액세스 타임이 25nsec인 SRAM으로 구현했다. 이 메모리의 크기는 192 Kbyte이다.

(3) 슬레이브 2단

슬레이브 2 프로세서는 중간값 생성과정을 가속화하는 KAFOG¹¹⁾를 사용하였다. KAFOG는 8Kword의 메모리를 액세스할 수 있는데, 이 시스템에서 사용되

는 300DPI(dots per inch)의 해상도에서는 2Kword 만으로도 KAFOG에서 출력되어 나오는 데이터를 모두 저장할 수 있으므로, 2Kword의 SRAM으로 구현하였다. 슬레이브 2프로세서로 이용되는 KAFOG의 동작원리 및 입력 데이터 형태는 아래와 같다.

i) 입출력 관계

(a) KAFOG의 입력 데이터가 line 명령어이면, 입력 line 데이터를 그대로 출력시킨다.

(b) KAFOG의 입력 데이터가 curve 명령어일 경우, 출력은 curve를 line segment로 근사화 시킨 line 데이터이다. 이 때 출력되는 line의 수는 KAFOG의 명령어와 함께 입력되는 값에 의해서 결정된다. dt데이터는 KAFOG를 문자 데이터에 대하여 계산하도록 동작시키기전에 초기화 시켜야할 파라메타이다.

ii) KAFOG입력 데이터 형태

KAFOG의 입력 데이터 형태와, 사용되는 명령어는 그림6과 같다.

(a) MOVE : line, curve 계산의 시작점을 나타낸다.

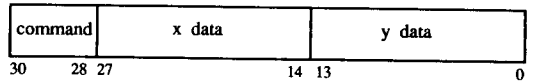
(b) curve(2,3,4차 point) : KAFOG는 3차 Bezier curve 계산을 H/W로 실현한 것이다. 이때 2차 3차 point는 Bezier curve식의 control point이고 4차 point는 곡선의 끝점이다.

(c) dt : curve를 몇개의 점선으로 나타낼 것인가를 결정하는 파라메타이다.

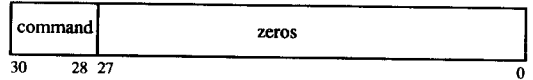
3. 출력단

연산단에 의해서 생성된 폰트의 비트 맵 데이터를

MOVE, LINE, CURVE



FONT START, FONT END



dt(parameter)

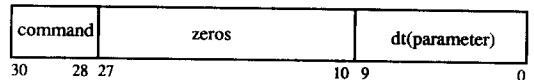


그림 6. KAFOG의 입력 데이터 형태

Fig. 6. Input data format of KAFOG.

은 출력단을 통해서 모니터와 LBP 엔진에 출력되며 출력단의 구성도는 그림 7에 나타난 것과 같다.

(1) 모니터의 출력회로

한글 그래픽 보드는 사용자에게 의해 요구된 외곽선 문자 발생 결과를 640×480의 해상도를 갖는 모니터로 나타내기 위한 비디오 출력 콘트롤 회로를 갖고 있다. 비디오 출력콘트롤 회로는 Frame buffer의 내용을 모니터에 나타내기 위해, 각 모니터의 화소에 해당되는 데이터와 비디오 타이밍 신호를 발생시킨다. 모니터의 각 화소는 0과 1의 값만을 갖는다. 즉 사용

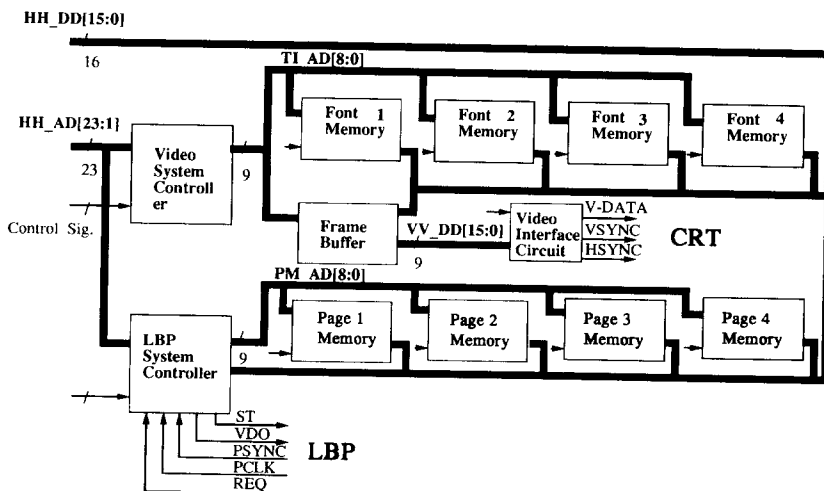


그림 7. 출력단의 블록 다이어그램

Fig. 7. Block diagram of output block.

되는 모니터는 모노크롬 모니터이다.

Frame buffer는 640×480의 해상도를 지원해 주기 위해서 64Kword의 VRAM으로 구현했다. 비디오 타이밍 신호는 TMS34061 VSC¹⁸에서 발생된다.

또한, TMS34061은 DRAM 제어 기능을 이용하여 4Mbyte의 폰트 메모리도 확장한다.

(2)LBP의 출력회로

300 DPI의 해상도를 갖는 TEC LBP 엔진을 이용하여 A4 용지를 기준으로하여 1쪽을 인쇄할 때 필요한 메모리의 크기는 2385×3390인데, 용지의 여백을 고려하면 8Mbit로써 1쪽에 대한 메모리 영역을 확보할 수 있다. 엔진으로의 출력 속도를 향상시키는 방법의 하나로 페이지 메모리 데이터를 LBP 엔진으로 보내는 동안 다른 페이지 메모리 단으로 데이터를 생성해서 보내는 더블버퍼링(double buffering) 기법을 사용하는데, 이를 위해서는 2배의 메모리 영역이 필요하다. 따라서 페이지 메모리의 크기는 2Mbyte가 되며, DRAM으로 구현된다.

LBP frame buffer의 데이터가 LPB 엔진으로 출력되는 과정을 다음과 같다. 먼저 마스터 프로세서가 LBP 엔진으로 데이터를 보낼 수 있는가를 확인한다. 그림 7의 REQ 신호가 LOW이면 엔진에게 데이터를 보낼 수 있으므로, REQ가 LOW일때 GP425¹⁹가 데이터 전송의 일을 할 수 있게 GP425의 내부 레지스터를 set한다. 또한, 마스터는 엔진의 ST 신호를 LOW로 하여 데이터 전송동작을 명령하는데, ST 신호가 LOW가 됨과 동시에 엔진의 PSYNC와 PCLK 신호가 구동된다. GP425는 LBP 인터페이스회로의 도움으로 LBP 엔진에서 나오는 PSYNC와 PCLK의 동기 맞추어 비디오 데이터인 VDO를 엔진으로 보낸다. 엔진으로 모든 데이터가 전송이되면 REQ신호는 다시 LOW가 되어서 다음의 데이터를 받을 준비를 한다.

IV. 실험 결과 및 고찰

제작된 KHGB90-B 시스템에 의해서 생성된 문자는 일차적으로는 640×480의 해상도를 갖는 모니터에 출력이 되고, 사용자의 요구가 있을 경우에는 2385×3390의 해상도를 갖는 LBP 엔진으로 출력 되어진다.

그림 8은 제작된 하드웨어의 사진이고, 그림 9는 이 시스템을 이용하여 생성한 문자를 모니터와 LBP에 출력한 것을 나타낸다. 지금 이 시스템은 32×32의 문자를 초당 약 100자 발생한다.

표1은 32×32 bit의 문자를 18,200자를 생성하여

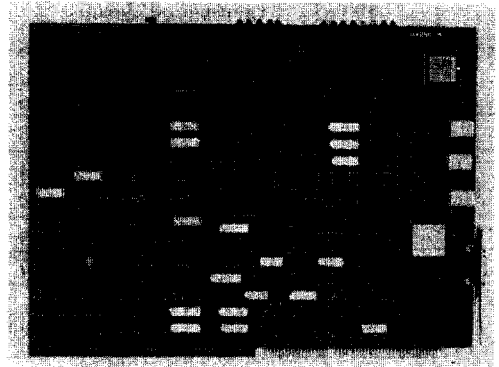
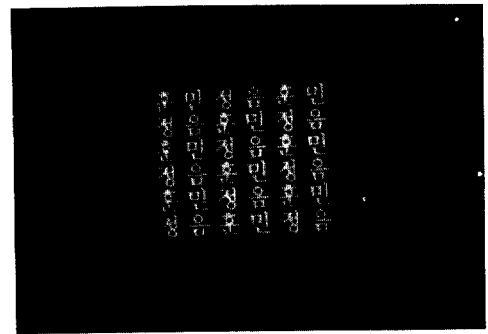
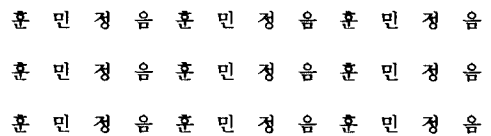


그림 8. 제작된 하드웨어의 사진
Fig. 8. The picture of the implemented hardware.



(a)



(b)

그림 9. 제작된 시스템으로 생성한 문자들
(a) 모니터에 출력된 문자들
(b) LBP에 출력된 문자들

Fig. 9. Characters generated by the system.
(a) Characters displayed on monitor,
(b) Characters displayed on LBP.

얻은 실험 결과 및 프로그램의 최적화(program optimization)후에 예측되는 성능을 나타낸 것이다. 주어진 수치는 문자생성에 요구되는 각각의 과정에서 소요되는 시간을 나타낸다. 최종 예측되는 값은 DSP 프로그램에서 내부의 멀티 데이터를 버스 고려하고, 마스터의 프로그램을 어셈블리어로 구현할 때의 값

표 1. 실험 결과

Table 1. Experimental results.

항 목	실 험 값	예 측 값
KAF0G	0.90	0.36
외곽선 생성	4.67	1.56
FILLING	3.33	1.11
DISPLAY	0.91	0.30
전체 소요 시간	10.00	3.34
발생 속도(CPS)	100	300

(단위 : msec)

으로서, 실험치의 1/3에 해당하는 값이다. 따라서, 최종 예측되는 글자 발생 속도는 약 300CPS가 된다.

또한, 앞으로 사용자들이 이 시스템을 보다 편리하게 사용할 수 있게 사용자 인터페이스 프로그램을 보완하고 각 프로세서의 프로그램을 최적화하여 시스템의 성능을 향상시키는 일들이 남은 과제가 되겠다.

參 考 文 獻

[1] 이태형, "외곽선 문자의 발생을 가속시키기 위한 VLSI 칩의 설계 및 구현" 한국과학 기술원 전기및 전자공학과 석사논문, 1990

[2] G.C. Hwang, Y.T. Lee, I.C. Park, T.H. Lee, J.H. Bae, and C.M. Kyung, "Hardware accelerator for scalable Hangul font generation," in JTC-CSCC, 1990, pp. 246-250

[3] PH. Coueignoux, "Character Generation by Computer," *Computer Graphics and Image Processing* 16, pp. 240-269, 1981.

[4] Roger D. Hersch, "Character Generation under Grid Constraints," *Computer Graphics*, vol. 21, Number 4, July, pp. 243-251, 1987.

[5] TMS34061 User's Manual, Texas Instrument, 1989.

[6] GP425 Data Sheet CIRRUS LOGIC, March, 1989.

著 者 紹 介

裴 鍾 洪 (準會員) 第28卷 B編 第9號 參照
현재 한국과학기술원 박사과정 재학중.

李 潤 泰 (正會員) 第28卷 B編 第9號 參照
현재 한국과학기술원 박사과정 재학중.

黃 奎 哲 (正會員) 第28卷 B編 第9號 參照
현재 한국과학기술원 박사과정 재학중.

慶 宗 旻 (正會員) 第28卷 B編 第9號 參照
현재 한국과학기술원 전기 및 전자공학과 교수