

전자 출판 시스템에 사용되는 고해상도 문자의 발생을 가속 시키기 위한 한글 그래픽 보드의 설계 및 제작

(Design and Implementation of Hanguk Graphic Board to speed up the
Generation of High Resolution Fonts used in Electric Public System)

黃 奎 哲*, 慶 宗 旻*

(Gyu Cheol Hwang and Chong Min Kyung)

要 約

본 논문은 한글이나 한자의 외곽선 정보로부터 점 행렬 문자 데이터를 발생시키기 위한 한글 그래픽 보드의 설계 및 제작에 관한 연구이다. 제작된 그래픽 보드는 TMS34010 그래픽 시스템 프로세서(GSP)와 TMS320C25 디지털 신호 처리 프로세서(DSP)로 구성되며, 두 프로세서의 병렬동작이 가능하도록 같은 주소를 같은 두개의 메모리 블록으로 구성된 공유 메모리를 두었다. 그리고 외곽선 정보로부터 점 행렬 문자 데이터를 계산하는데 많은 시간을 필요하는 Bezier curve 계산을 DSP 환경에서 효율적으로 수행 시키기 위한 알고리즘을 제안한다.

Abstract

In this thesis, we represent the study on the design and implementation of the hanguk graphic board which generate bit map font data from the boundary information of korean or chines fonts.

The implemented graphic board consists of a TMS34010 Graphic System Processor (GSP) and a TMS320C25 Digital Signal Processor (DSP), and there is shared memory which consists of two memory blocks with same address for which is possible parallel processing between two processors. And in using DSP, we propose an efficient algorithm for calculation of Bezier curve which require much times to calculate bit map data font from the boundary information.

I. 서 론

컴퓨터의 속도가 빨라지고 기억용량이 급격히 증대함에 따라 컴퓨터 응용분야로 Desk-top publishing 시스템이나 LBP (laser beam printer)를 이용한 전자출판 시스템에 대한 관심이 높아지고 있다. 전자출판 시스템이란, 일반적인 출판업무나 사무용 문서작

성, 신문편집과 같은 일을 고해상도의 Workstation이나 LBP와 같은 전자시스템을 이용함으로써, 종래의 기계적 방식만을 사용하던 출판업무에서 할 수 없었던 양질의 다양한 문자의 발생을 가능하게 하여 보다 편리한 출판업무를 수행하게 한다. 컴퓨터를 이용한 전자출판 시스템에서 다양한 문자의 발생을 위해서는, 문자에 대한 정보를 컴퓨터에 저장하는 문자 부호화 방법과, 저장된 문자 부호로부터 원하는 출력장치로 문자를 발생시키는 과정이 매우 중요하다.

일반적으로 컴퓨터에서 글자체 (FONT)를 표현하는 방법으로 점행렬 글자체 (Bitmap Font), 외곽선 글

*正會員, 韓國科學技術院 電氣 및 電子工學科
(Dept. of Elec. Eng., KAIST)
接受日字: 1989年 12月 6日

자체 (Countour or Boundary coding Font), 구조적 글자체 (Structural coding Font)가 있다. 이들 방법의 선택은 출력장치의 특성, 응용 소프트 및 사용자의 요구조건에 따라 결정된다.

점 행렬 글자체는 컴퓨터의 단말장치나 dot-matrix 프린터에 글자를 나타내기 위해 가장 많이 사용되는 방법으로 출력 속도는 빠르지만, 고해상도의 출력기기를 사용하거나, 사용자가 요구하는 문자의 형태가 다양한 전자출판 시스템의 경우 사용되는 모든 문자의 점 행렬 문자 데이터를 컴퓨터가 저장해야 함으로 많은 기억 장소를 필요로 한다.

외곽선 글자체나 구조적 글자체의 경우, 컴퓨터가 갖는 글자체의 정보는 글자체 형태의 외곽선 점들의 집합을 직선, 원, 자유곡선 등으로 표현하고 있어 사용하는 출력장치의 해상도에 무관하게, 그리고 사용자가 원하는 글자의 형태를 쉽게 만들 수 있다. 현재 많이 사용되고 있는 Postscript, Impress 등의 Page Description Language (PLD)와 여러 LBP에서 외곽선 코딩 방법을 이용하여 영문자나 숫자, 특수 기호들에 대해서 만족할 만한 모양의 글자체를 출력하고 있다. 그러나 글자의 외곽선 정보로부터 점 행렬 문자 데이터를 만들기 위해 산술 연산과 그래픽에 관련된 연산 (line draw, filling, pixel block transfer...)을 수행하여야 함으로 한글과 같이 글자의 종류가 많고 형태가 다양한 글자체의 경우 많은 시간이 요구된다.

본 논문은 전자 출판 시스템에서 외곽선 정보로 주어진 한글이나 한자로 부터 점 행렬 문자 데이터를 고속으로 계산하기 위한 그래픽 가속장치의 개발에 대한 연구이다.

II. 한글 그래픽 보드

1. 한글 그래픽 보드의 동작 환경

본 논문에서 제작한 한글 그래픽 보드는 IBM-PC AT의 add on 보드로 동작한다. 보드상에서 동작하는 응용 소프트웨어는 C 프로그래밍 언어나 TI (texas instruments)에서 제공하는 TMS34010 GSP (graphic system processor)와 TMS320C25 DSP (digital signal processor)의 어셈블리 언어로 IBM-PC에서 프로그램 된다. 그리고 이 프로그램은 PC 상에서 Cross C 컴파일러와 Cross 어셈블러에 의해 실행 파일로 된다. 이 실행 파일과 PC에서 편집된 데이터를 down loading 프로그램에 의해 한글 그래픽 보드의 주 메모리로 옮겨져 수행된다. 그리고 한글 그래픽 보드에서 수행되는 문서 편집 데이터 중에서 화면에 나타내어질 글자는 외곽선 정보로 주어진 문

자이므로, 점 행렬 글자로 변환이 필요하여, 이를 위한 계산이 한글 그래픽 보드에서 요구된다. 한글 그래픽 보드는 계산한 점 행렬 문자를 640×480를 갖는 다중 동기 모니터로 출력한다.

그림 1은 한글 그래픽 보드의 환경을 나타낸 것이다.

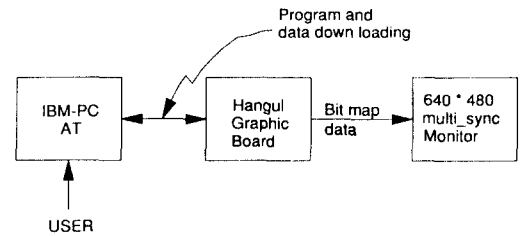


그림 1. 한글 그래픽 시스템의 주변 환경
Fig. 1. Enviornment of the hangul graphic system.

2. 한글 그래픽 보드의 구성

제작된 한글 그래픽 보드는 TMS34010 그래픽 프로세서와 TMS320C25 디지털 신호처리 프로세서, 그리고 두 프로세서가 독립적으로 일을 수행할 수 있게 하기 위해 같은 주소 영역을 갖는 두개의 메모리 블록으로 구성된 공유 메모리로 구성된다. 그림 2는 한글 그래픽 보드의 전체 블록 다이어그램을 나타낸다.

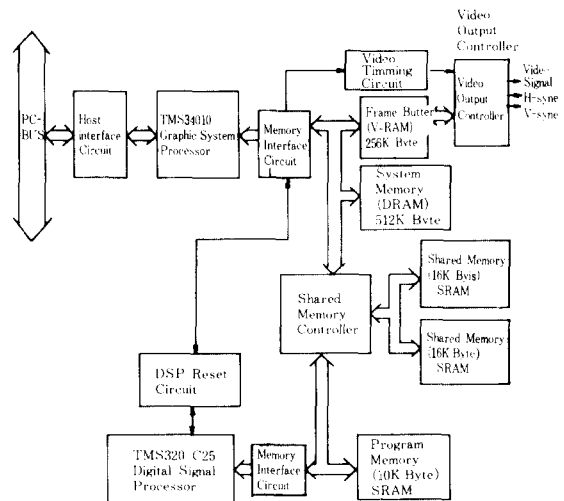


그림 2. 한글 그래픽 보드의 블록 다이어그램
Fig. 2. Block diagram of the hangul graphic board.

여기서 그래픽 시스템 프로세서와 디지털 프로세서 각각이 수행해야 할 일은 다음 절에서 자세히 설명된다.

1) TMS34010 그래픽 프로세서의 응용

TMS34010 그래픽 프로세서는 한글 그래픽 보드의 주 프로세서로 동작한다. 이 프로세서가 하는 일을 분석하면 아래와 같이 요약할 수 있다.

- (i) IBM-PC 와의 통신을 위한 PC-interface 회로 제어
- (ii) 다중 동기 모니터를 동작 시키기 위한 수직 및 수평동기 신호 발생
- (iii) CRT 및 DRAM refresh 신호 및 어드레스 신호의 발생
- (iv) TMS320C25 디지털 신호 처리 프로세서를 문자의 외곽선 정보로부터 점 행렬 문자 데이터를 발생 시키기 위한 보조 프로세서로 동작 시키기 위한 제어 프로그램 수행
- (v) IBM-PC로부터 온 문자 코드의 해석 및 일반적인 그래픽 기능의 수행 (ex:line, circle, filling, pixel block transfer...)

2) TMS320C25 디지털 신호 처리 프로세서의 응용

TMS320C25 디지털 신호 처리 프로세서는 그래픽 시스템 프로세서에서 요구된 외곽선 문자 데이터로부터 점 행렬 문자 데이터를 구하기 위해 요구된 일을 수행하기 위해 사용되는 점 행렬 문자 계산용 보조 프로세서로 사용된다.

외곽선 문자 정보로부터 점 행렬 문자 데이터를 만들기 위해 필요한 연산은 아래와 같이 분류된다.

(1) 좌표 변환 및 scaling

$$[x' \ y'] = [x \ y] \times \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} + [dx \ dy]$$

(2) rotation

$$[x' \ y'] = [x \ y] \times \begin{bmatrix} \sin\theta & \cos\theta \\ -\cos\theta & \sin\theta \end{bmatrix}$$

(3) curve 계산(3차 bezier curve 계산)

$$X(t) = X_0 \times (1-t)^3 + 3 \times X_1 \times (1-t)^2 \times t + 3 \times X_2 \times (1-t) \times t^2 + X_3 \times t^3$$

$$Y(t) = Y_0 \times (1-t)^3 + 3 \times Y_1 \times (1-t)^2 \times t + 3 \times Y_2 \times (1-t) \times t^2 + Y_3 \times t^3$$

- x, y : 고유의 font 데이터(좌표값)
- x', y' : 변환 후의 좌표값
- X₁, Y₁ : 3차 Bezier curve의 특이점
- X[t] : 계산된 curve 데이터 t=0 ≤ t ≤ 1

이들과 더불어 점 행렬 문자 계산을 위해 필요한 그

래픽 연산으로 임의의 다각형 내부를 채우는 Filling 알고리즘이 있는데 이는 참고문헌^{[1][2][3]}을 참조하기 바란다.

3) 공유 메모리의 구조 및 제어 방법

공유 메모리는 두 프로세서(GSP와 DSP) 사이의 병렬동작을 최대한 보장하기 위해 같은 주소 영역을 갖는 두개의 메모리 블록으로 구성된다. 그림3은 공유 메모리의 블록 다이어그램을 보여준다.

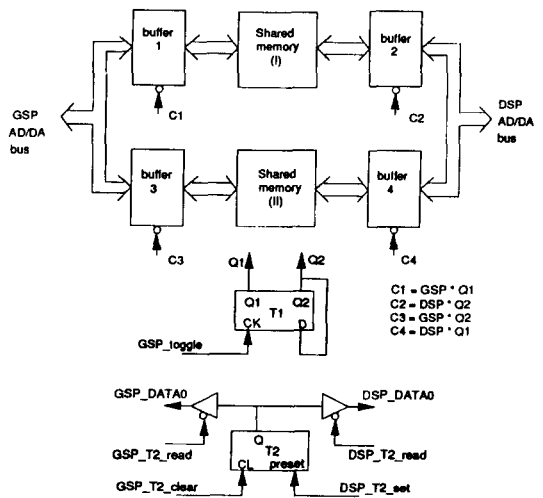


그림 3. 공유 메모리의 블록 다이어그램
Fig. 3. Block diagram of the shared memory.

그림에서 메모리 블록(I)과 메모리 블록(II)는 GSP와 DSP 각각의 어드레스 영역에서 같은 주소에 해당된다. 만약 메모리 블록(I)이 GSP로 부터 접근 가능하다면 DSP는 반드시 메모리 블록(II)에 접근하도록 버퍼 제어신호 C₁, C₂, C₃, C₄를 발생시킨다. GSP와 DSP가 다른 메모리 블록에 접근하는 것은 GSP에서 발생하는 toggle 신호에 의해 제어된다. 다음은 GSP와 DSP를 효율적으로 이용하기 위한 공유 메모리 통신 규약이다.

- (i) T₂의 초기 상태는 시스템 reset에 의해 "1"이 된다.
- (ii) T₂의 출력 Q가 "1"이면 GSP와 DSP은 아래의 동작을 수행할 수 있다.
GSP : 공유 메모리에 데이터를 쓰고 다음 T₁를 toggle 시킨다. (Q₂=0) 그리고 공유 메모리로부터 바로 이전에 DSP에 의해 계산된 데이터를 읽어 그래픽 연산을 수행한다.
DSP : Q₂가 0이 될때까지 기다린다.

(iii) T_2 의 출력 $Q_2=0$ 일 경우

GSP : 공유 메모리에 데이터를 쓰고 $Q_2=1$ 이 될 때까지 기다린다.

DSP : 공유 메모리의 데이터에 대해서 필요한 연산을 수행하고 결과를 다시 공유 메모리에 쓴다.

(ii)에서 $Q_2=1$ 일 때 GSP에서 DSP에 요구할 산술 연산의 입력 데이터를 공유 메모리에 쓰고 T_1 을 GS GSP-toggle 신호를 이용해 toggle 시킴과 동시에 T_2 를 clear 시킨다. 이때 T_2 의 출력 Q_2 가 0이 되므로 통신 규약에 의해 DSP는 GSP에 요구된 산술연산을 DSP와 연결된 공유 메모리에 있는 데이터에 대해 수행하고, GSP는 바로 이전에 DSP에 의해 계산된 결과를 GSP와 연결된 공유 메모리로 부터 읽어 그래픽 연산을 수행할 수 있어 GSP와 DSP의 병렬 동작이 가능해진다.

만약 DSP의 연산수행 속도가 GSP의 그래픽 연산 수행속도 보다 빠르면 GSP에서 느끼는 DSP의 점행렬 문자 연산 수행속도는 GSP가 공유 메모리에서 데이터를 접근하는 시간과 같게 된다.

3. 한글 그래픽 보드의 GSP와 DSP에 각각에 대한 제어 루틴

한글 그래픽 보드의 원활한 동작을 가능케 하기 위한 GSP와 DSP의 제어 루틴은 그림 4와 같다.

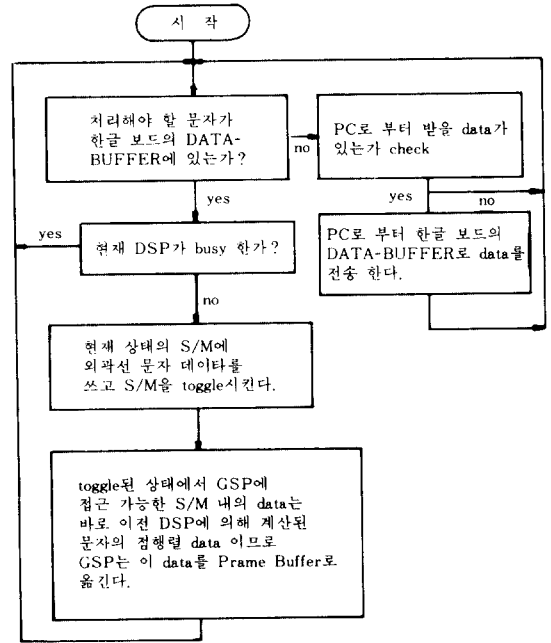
그림 4의 제어 흐름도는 GSP와 DSP 그리고 공유 메모리를 제어 하기위한 제어 루틴을 나타낸 것이다. 그림 4(a)의 GSP 제어 흐름도는 PC와 한글 그래픽 보드와의 데이터 교환을 위한 통신 루틴, PC에서 편집된 데이터를 보드의 모니터에 나타내기 위해 필요한 제어 루틴 그리고 외곽선 문자로 부터 점행렬 문자를 발생시키기 위해 DSP와 공유 메모리를 제어하는 부분이 포함된다.

그림 4(b)의 DSP 제어 흐름도는 GSP에서 점행렬계산이 요구된 문자 계산을 위한 일과 그리고 GSP와 보조 메모리 사용을 위한 제어 루틴이 포함된다. 그리고 점행렬 문자 계산을 위해 DSP에서 수행되어야 할 연산은 다음 III장에서 자세히 설명된다.

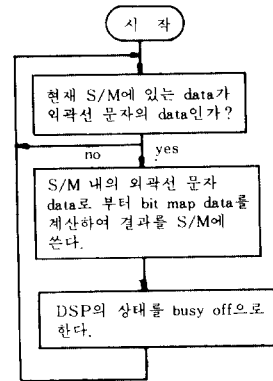
III. DSP를 이용한 3차 Bezier Curve 계산 알고리즘

DSP 환경에서 매개변수를 이용한 3차 Bezier curve 식으로부터 곡선을 이루는 값을 계산하기 위한 방법을 제안한다.

먼저 DSP에서 쉽게 처리될 수 있는 연산은 $A \times B + C$ 의 형태의 계산식이고 이와같은 계산은 DSP에



(a)



(b)

그림 4. (a) GSP 제어 흐름도 (b) DSP 제어 흐름도

Fig. 4. (a) GSP control flow chart, (b) DSP control flow chart.

서 한번의 명령어로 수행 가능하다.

본 논문에서 제안한 방법은 2차 Bezier curve 식을 위 형태의 식으로 변환시켜 후 곡선값을 계산한다. 그리고 Bezier curve 식에서 곡선 값을 계산하는 방법은 곡선을 직선의 연결로 모델링한 선형분할 방법을 이용한다.^{[3][4][5][6]}

X, Y 좌표계에서 X에 대한 3차 Bezier curve 식은 다음과 같다.

$$X(t) = X_0 \times (1-t)^3 + 3 \times X_1 \times (1-t)^2 \times t + 3 \times X_2 \times (1-t) \times t^2 + X_3 \times t^3$$

X_i : 곡선의 특이점, $t=0, 1, 2, 3$

$$0 \leq t \leq 1$$

먼저 식 1을 아래와 같이 변형 시킬 수 있다.

$$X(t) = A \times t^3 + B \times t^2 + C \times t + D$$

$$= ((A \times t + B) \times t + C) \times t + D$$

여기서

$$A = 3(X_1 - X_2) + 3X_3 - X_0$$

$$B = 3(X_0 - 2X_1 + X_2)$$

$$C = 3(X_1 - X_0)$$

$$D = X_0$$

이다.

식 2와 같이 변형된 식은 $A \times B + C$ 의 recursive한 형태이므로 DSP 환경에서 효율적으로 사용할 수 있다.

다음은 식 1과 식 2 각각을 계산하기 위해 DSP가 수행 해야할 instruction을 근사적으로 계산한 것이다. 먼저 3차 Bezier curve 식의 특이점들이 DSP 내부 메모리에 이미 존재한다고 가정한다. (이 경우 명령어 수행에 필요한 데이터를 읽어오는 시간을 무시할 수 있다)

1식의 경우 12번의 명령어를 수행후 1개의 곡선 데이터를 구할 수 있는 반면, 식 2의 경우 3번의 명령어 수행후에 한개의 곡선 데이터를 구할 수 있어 초기에 계산해야할 A, B, C, D 계산시간을 무시한다면 식 2는 식 1에 비해 DSP에서 약 4 배의 속도로 계산된다. 여기서 명령어는 DSP에서 $A \times B$ 와 $A \times B + C$ 형태를 수행할 수 있는 instruction을 말한다.

IV. 결 과

그림 5와 그림 6는 본 연구에서 제작된 한글 그래픽 보드를 이용하여 한글 및 한자의 외곽선 정보로부터 640×480 해상도를 갖는 다중 동기 모니터에 256×256 크기로 나타낸 것이다.

그림 7은 32×32 크기의 한글과 한자를 동시에 한 화면에 나타낸 예이다.

한글 그래픽 보드의 동작 속도는 그림 7과 같은 32×32 크기의 글자를 초당 50사 정도 화면에 나타낼 수 있다.

V. 결 론

본 논문에서는 외곽선 정보로부터 점 행렬 문자를 빠른 속도로 계산하기 위한 한글 그래픽 보드의 제작에 관한 연구이다. 제작된 한글 그래픽 보드는



그림 5. 256×256 크기의 한글
Fig. 5. Korean fonts with 256×256 size.



그림 6. 256×256 크기의 한자
Fig. 6. Chinese fonts with 256×256 size.

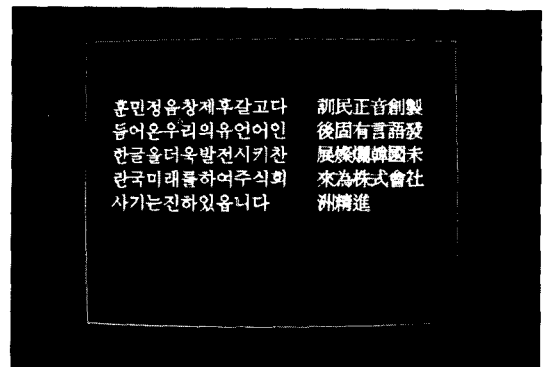


그림 7. 32×32 크기의 한글 및 한자
Fig. 7. Korean and chinese fonts with 32×32 size.

TMS34010 GSP와 TMS320C25 DSP로 구성되며, 두 프로세서의 병렬 동작이 가능하도록 하기 위한 2중 보조 메모리를 갖고 있다. 현재까지 개발된 보드의 응용 소프트웨어로 보드의 문자 발생 속도는 32×32 크기의 문자에 대해 초당 약 50 문자를 계산할 수 있다. 앞으로 임의의 폐곡선 내부를 채우는 Filling에 대한 DSP 환경에서의 알고리즘 개발과, 보드를 효율적으로 운용하는 소프트웨어를 개발한다면 보다 높은 계산 속도를 갖을 수 있으리라 생각된다.

參 考 文 獻

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

[2] Roger D. Hersch, "Character generation

under grid constraints," *Computer Graphics*, vol. 21, no. 4, pp. 243-251, July 1987.

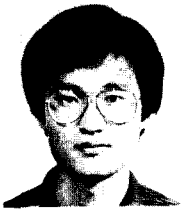
[3] J.D. Foley and A. Van Dam, "Fundamentals of interactive computer graphics," *Adisson Wesley*, 1984.

[4] Wolfgang Bohm, Gerald Farin, Jurgen Kanmann, "A survey of curve and surface methods in CAGD," *Computer Aided Geometric Design* 1, pp. 1-60, 1984.

[5] William M. Newman, Robert F. Sproull, "Principal of interactive computer graphics," *McGraw Hill*, second edition, 1986.

[6] Jerry Van Aken, Mark Novak, "Curve drawing algrotihm for raster displays," *ACM Tran. on Graphics*, vol. 4, no. 2, pp. 147-169, April 1985.

著 者 紹 介



黃 奎 哲 (正會員)

1964年 9月 16日生. 1983年 3月 ~1987年 2月 경북대학교 전자공학과 학사학위 취득. 1987年 3月 ~1989年 2月 한국과학기술원 전기및 전자공학과 석사학위 취득. 1989年 3月 ~현재 한국과학 기술

원 전기및 전자공학과 박사 과정. 주관심분야는 전자출판 시스템과 컴퓨터 그래픽 시스템의 Hardware 구조 연구 등임.

慶 宗 旻 (正會員) 第25卷 第10號 參照

현재 한국과학기술원 전기및 전자공학과 부교수