

(19) 대한민국특허청(KR)
(12) 특허공보(B1)

(51) Int. Cl. ⁶ G06F 15/62		(45) 공고일자 1997년03월 17일	
		(11) 공고번호 특1997-0003326	
		(24) 등록일자 1997년03월 17일	
(21) 출원번호	특1993-0003001	(65) 공개번호	특1993-0022226
(22) 출원일자	1993년03월02일	(43) 공개일자	1993년11월23일
(30) 우선권주장	92-080055 1992년04월01일 일본(JP)		
(73) 특허권자	가부시끼가이샤 히다찌세이사꾸쇼 가나이 쓰토무		
(72) 발명자	일본국 도쿄도 지요다구 간다 스루가다이 4-6 와다 히로시 일본국 후지사와의 쓰지도 5981 기따즈메 요시아끼 일본국 사야마시 가미오꾸또미 19-18 하세가와 가즈꼬 일본국 요코하마시 이즈미구 나까따초 1056-1 와끼사카 신지 일본국 요코하마시 도쓰까구 도쓰까초 1745 사또 히로꼬 일본국 요코하마시 도쓰까구 요시다초 1545 미야모토 다카시 일본국 도쿄도 스기나미구 오기꾸보 1-3-5 백남기		
(74) 대리인	백남기		

심사관 : 박제현 (책자공보 제4878호)

(54) 문자 도형 발생 회로

요약

내용없음.

대표도

도1

명세서

[발명의 명칭]

문자 도형 발생 회로

[도면의 간단한 설명]

제1도는 본 발명의 1실시예의 구성의 설명도.

제2도는 레지스터와 연산기의 복합 종속 접속 기구를 도시한 설명도.

제3도는 데이터 변환회로와 동작의 알고리즘을 나타낸 흐름도.

제4도는 아우트 라인 폰트 전개 처리의 설명도.

제5도는 아우트 라인 폰트 전개 처리 시간예의 설명도,

제6도는 프로그램 ROM의 내용을 외부에서 갱신하는 방법을 나타낸 설명도.

제7도는 본 발명의 다른 실시예의 구성의 설명도.

제8도는 데이터 변환회로의 동작의 알고리즘을 나타낸 흐름도.

제9도는 레스터 변환회로의 구성예의 설명도.

제10도는 레스터 변환회로의 동작의 알고리즘을 나타낸 흐름도.

제11도는 제7도의 실시예를 변형한 또 다른 실시예의 구성의 설명도.

제12도는 제1도의 실시예를 변형한 다른 실시예의 구성의 설명도.

제13도는 제7도의 실시예를 또 변형한 다른 실시예의 구성의 설명도.

제14도는 2개의 도형 발생 LSI를 갖는 다른 실시예의 구성의 설명도.

제15도는 제7도의 실시예를 또 변형한 다른 실시예의 설명도.

[발명의 상세한 설명]

본 발명은 문자 도형 작성 장치 등의 정보처리 장치에 있어서 벡터 형식으로 표시된 윤곽 정보에서 도트 형식의 문자 도형 데이터를 발생하는 문자 도형 발생 회로 및 시스템에 관한 것이다.

나가시마 등의 'DESIGN OF AN OUTLINE FONT RASTERIZING LSI' (IEEE CUSTOM INTEGRATED CIRCUITS CONFERENCE 9-2 pp. 117-119. 1989)에는 곡선보간, 문자나 도형의 윤곽선 발생, 윤곽선 내부의 페인팅을 위한 LSI의 설계가 개시되어 있다.

본 발명의 목적은 선택된 타입의 폰트 데이터 입력에 대응해서 윤곽선 데이터에서 도트 데이터로 변환하는 프로그램을 갖는 메모리를 구비한 데이터 발생용 LSI 회로를 제공하는 것이다.

본 발명의 다른 목적은 벡터 형식의 문자나 도형의 윤곽 정보에서 도트 형식의 문자 도형 데이터를 발생하는 일련의 처리를 손상하는 일없이 회로 규모 또는 처리 회로의 구성을 축소한 문자 도형 발생 회로를 제공하는 것이다.

본 발명의 다른 목적은 윤곽선 발생 등의 처리량이 많은 처리에 대해서는 전용의 회로를 사용하여 고속으로 실행하는 것에 의해, 작은 회로 규모로 고속이며 또한 효율적으로 도트 형식의 문자 도형 데이터를 발생할 수 있는 도트 형식에 데이터 발생 회로를 제공하는 것이다.

본 발명의 또 다른 목적은 프로그램 ROM 등의 내용을 자유롭게 변경하는 것이 가능하게 되고, 벡터 형식의 문자 도형 데이터가 각종의 포맷을 채용하고 있더라도 대응 가능한 도트 형식 데이터 발생 회로를 제공하는 것이다.

상기 목적을 달성하기 위한 문자 도형 발생 회로로서 윤곽 데이터를 비트 맵 데이터로 변환하는 회로 블록은 폰트 데이터를 소정의 파라미터에 따라 변환(신장/좌표 변환, 곡선보간)하기 위한 여러개의 명령을 기억하기 위한 메모리, 상기 메모리에 접속되고 폰트 데이터에 상기 메모리에 기억된 여러개의 명령에 의해서 지시되는 변환을 실시하기 위한 제1데이터 변환회로 수단, 상기 레지스터와 상기 연산기에 접속되고 상기 변환을 실시한 데이터를 순차적으로 받아들여서 유지하고 그 데이터를 받아들인 순서대로 출력하는 선입 선출 수단, 상기 선입 선출 수단에 접속되고 상기 선입 선출 수단에서 순서대로 출력하는 해당 데이터 비트 맵 데이터로 변환하는 제2데이터 변환 회로 수단, 상기 제2데이터 변환 회로 수단에 접속되고 상기 비트 맵 데이터를 유지하는 비트 맵 메모리, 상기 제2데이터 변환 회로 수단은 해당 데이터를 래스터화하기 위한 래스터 변환 실행 유닛, 상기 제2데이터 변환 회로 수단은 상기 윤곽선 내부 부분의 페인트 종료후 페인트 종료신호를 발생하는 종료신호 발생 수단을 포함하고, 상기 제1데이터 변환 회로 수단은 상기 소정의 파라미터를 기억하는 레지스터와 상기 폰트 데이터를 연산하기 위한 연산기를 갖고, 상기 제1데이터 변환 회로 수단은 상기 선입 선출 수단이 유지 데이터로 가득찬 상태인 경우는 데이터 변환하는 것을 중단하고 빈곳에 생기는 것을 대기하고, 상기 제2데이터 변환 회로 수단은 상기 선입 선출 수단이 데이터가 빈 상태이면 데이터가 그 속에 유지될 때까지 대기하고, 상기 래스터 변환 실행 유닛을 제어하는 콘트롤러를 갖고 상기 비트 맵 메모리 중의 변환된 비트맵 데이터의 윤곽선 내부 부분의 페인트를 실행하며, 상기 회로 블록은 하나의 대규모 집적회로내에 구성되고 상기 메모리는 프로그램 가능한 ROM인 것을 특징으로 한다.

벡터 형식으로 표시된 문자나 도형의 윤곽 정보에서 도트 형식의 문자 도형 데이터를 발생하는 문자 도형 발생회로에 있어서, 각조의 명령을 기억하는 제1의 기억영역, 상기 제1의 기억영역에 기억되어 있는 명령을 순차 해석하고, 윤곽 정보에 목적의 문자를 작성하기 위한 변환을 가하는 제2의 논리 유닛, 그 변환이 가해진 윤곽 정보에 또 도트 형식의 데이터로서의 변환을 가하는 제2의 논리 유닛 및 도트 형식의 문자 도형 데이터를 일시적으로 기억하는 제2의 기억영역을 동일 LSI 기관상에 배치한 구성을 갖는 회로는 폰트 데이터 변환을 고속으로 실시한다.

또, 제1의 기억영역으로서 라이트 가능한 기억수단을 사용하고, 상기 기억수단은 문자 도형 발생회로의 외부에서 필요한 데이터의 라이트를 가능하게 한 구성도 고려된다. 또, 제1의 기억영역으로서 RAM을 사용한 구성이어도 좋다.

우선, 제1의 기억영역에 벡터 형식으로 표시된 문자 도형의 윤곽 정보에서 도트 형식의 문자 도형 데이터를 발생시키기 위해서 필요한 처리의 일부를 나타내는 감산, 가산, 승산, 베지에르 커브(Bezier curve) 발생의 마이크로 명령을 사전에 기억해 둔다.

다음에 제1의 논리 유닛은 벡터 형식의 윤곽정보를 받아들였을때 제1의 기억영역에 기억된 명령을 해석해서 윤곽정보에 미리 사용자에게 의해 지정되어 레지스터에 유지된 배율에 따라서 확대/축소 등의 변환을 가한다. 이러한 변화는 예를 들면 데이터 신장, 좌표 변환, 각종의 보정처리, 곡선보간 등을 포함한다. 상기 변환 후의 윤곽 정보는 제2의 논리 유닛에 전송된다. 제2의 논리 유닛은 전송된 윤곽 정보에 또 변환을 가하여 도트 형식의 문자 도형 데이터를 제2의 기억영역에서 발생시킨다.

이상과 같이, 제2의 논리 유닛은 제1의 논리 유닛에서 받아들인 각종 변환후의 윤곽정보에서 비트 맵 형식의 문자 도형의 윤곽선 정보를 제2의 기억영역내에 발생시키고, 윤곽선 내부의 페인팅 등의 처리를 실행한다.

이와 같은 처리를 실행하는 제1의 논리 유닛, 제2의 논리 유닛, 제1의 기억영역 및 제2의 기억영역을 동일기관상에 배치하는 것에 의해, 소형이고 고속처리가 가능한 도형 발생회로를 제공하는 것이 가능하게 된다.

다음에 본 발명의 1실시예를 제1도에 따라서 설명한다.

제1도의 워드 프로세서, 퍼스널 컴퓨터, 프린터 등의 각종 정보처리 장치에 있어서 사용되는 본 발명에 관계된 문자 도형 발생회로의 구성예를 구성예를 도시한 것이다.

본 실시예의 문자 도형 발생회로(4)는 CPU(1), 폰트 RAM(2), 출력 엔진(3)에 버스(18)를 거쳐서 접속된다. 여기에서, CPU(1)는 문자 도형 발생회로, 인쇄, 표시 등의 각종의 처리의 제어를 실행하는 중앙처리 장치로서, 반도체 마이크로 프로세서 등으로 구성된다. 폰트 RAM(2)은 벡터 형식으로 표시된 문자 도형 데이터를 저장하는 수단으로서, 반도체 IC 등의 전자 디바이스로 구성된다. 출력 엔진(3)은 도트 형식으로 표시된 문자 도형 데이터를, 예를 들면 프린터 등에 의해 인쇄 등을 하기 위해 상기 프린터를 구동하기 위한 수단으로서 각종 TTL, 코일, 파워 트랜지스터 등의 전자 디바이스로 구성된다.

문자 도형 발생회로(4)는 프로그램 ROM(5), 데이터 변환회로(6), FIFO(16), 래스터 변환회로(13)를 갖는다.

프로그램 ROM(5)은 벡터 형식의 문자 도형 데이터에 데이터 신장, 좌표 변환, 보정, 곡선 보간 등의 처리를 실행하는 프로그램을 저장하는 수단으로서, 반도체 IC 등의 전자 디바이스로 구성된다.

데이터 변환 회로(6)은 데이터 변환 컨트롤부(7) 및 데이터 변화 실행 유닛(10)(승산기, 가산기 등)을 갖고, 프로그램 ROM(5)의 내용을 리드 및 디코드하며, 벡터 형식의 문자 도형 데이터에 각종의 변환을 가하는 수단이다.

또, 데이터 변환 컨트롤부(7)는 명령 디코더(8) 및 프로그램 카운터 PC(9)를 갖고 구성되며, 데이터 변환 회로(6)에 있어서 데이터 변환 실행 유닛(10)을 제어하는 수단이다.

또, 명령 디코더(8)는 프로그램 ROM(5)에서 리드한 내용을 디코드하는 명령 디코드 수단으로서, 각종 TTL, 반도체 메모리 등의 전자 디바이스로 구성된다.

또, 프로그램 카운터 PC(9)는 상술한 마이크로 명령을 갖는 프로그램 ROM(5)의 리드 어드레스를 제어하는 수단으로서, 각종 TTL, 반도체 메모리 등의 전자 디바이스로 구성된다.

또, 데이터 변환 실행 유닛(10)은 레지스터군(11) 및 연산기(12)를 갖고, 데이터 변환 처리를 실행하기 위해 필요한 각종 연산을 실행하는 수단이다.

또, 레지스터군(11)은 각종 명령, 처리 결과 등을 일시적으로 기억해 두는 수단으로서, 각종 TTL, 반도체 메모리 등의 전자 디바이스로 구성된다.

또, 연산기(12)는 가산, 감산 등의 각종 연산을 실행하는 수단으로서, 각종 TTL, 반도체 메모리 등의 전자 디바이스로 구성된다.

다음에 FIFO(16)는 소위 선입 선출 기억수단으로서, 반도체 메모리 등의 전자 디바이스로 구성된다.

래스터 변환 회로(13)는 컨트롤부(14) 및 래스터 변환 실행 유닛(15)을 갖고 구성되고, 데이터 변환회로(6)에 의해서 변환이 가해진 데이터에서 도트 형식의 문자 도형 데이터를 발생하는 수단으로서, 실제로는 각종 TTL, 반도체 메모리 등의 전자 디바이스로 구성된다.

또, 컨트롤부(14)는 래스터 변환을 위한 실행 유닛(15)을 제어하는 수단으로서, 각종 TTL, 반도체 메모리 등의 전자 디바이스로 구성된다.

또, 래스터 변환 실행 유닛(15)은 래스터 변환을 실행하기 위해 필요한 각종 연산을 실행하는 수단으로서, 각종 TTL, 반도체 메모리 등의 전자 디바이스로 구성된다.

워크 메모리(17)는 도트 형식의 문자 도형의 데이터를 일시적으로 저장하는 수단으로서, 반도체 메모리 등의 전자 디바이스로 구성된다.

상술한 본 발명의 장치 구성을 사용해서 문자 도형을 출력하는 순서에 의해서 설명한다.

우선, CPU(1)는 CPU상에서 런하는 프로그램에서의 지시에 따라 폰트 ROM(2)에 저장되어 있는 벡터 형식의 어떤 윤곽선 정보, 즉 문자 도형 데이터를 문자 도형 발생회로(4)에 전송한다.

문자 도형 발생회로(4)내의 데이터 변환회로(6)는 프로그램 ROM(5)에 저장된 명령군을 PC(9)에 저장되어 있는 값에 따라서 리드하고, 리드한 명령을 명령 디코더(8)에 의해서 디코드한다.

디코드된 명령을 데이터 변환 실행 유닛(10)내의 레지스터군(11) 및 연산기(12)를 사용해서 실행한다. 실행종료후 변환을 가한 데이터를 FIFO(16)에 기억한다.

래스터 변환회로(13)는 래스터 변환 컨트롤부(14) 및 래스터 변환 실행 유닛(15)을 사용해서 FIFO(16)에서 리드한 데이터에서 문자 도형의 윤곽선을 워크 메모리(17)상에 발생하고, 또 워크 메모리(17)상의 윤곽선의 내부를 페인팅한다.

마지막으로 CPU(1)가 워크 메모리(17)상에 발생된 도트 형식의 문자 도형 데이터를 리드하여 출력 엔진(3)으로 전송하고, 인쇄 및 표시 등을 실행하는 것이 가능하게 된다.

여기에서, 데이터 변환회로(6)와 래스터 변환회로(13) 사이에 FIFO(16)를 마련한 이유는 다음과 같다.

데이터 변환회로(6)와 래스터 변환회로(13)는 서로 독립적으로, 즉 비동기로 동작한다. 이 때문에, 데이터 전송에 있어서 동기를 취하기 위해서는 중간에 FIFO(16)를 마련하여 제어할 필요가 있다. 물론, 데이터 변환회로(6)가 FIFO(16)에 데이터를 기억할 때에 이미 FIFO(16)의 기억영역이 풀 상태이면 기억시키지 않고 FIFO(16)의 기억영역에 빈 영역이 생길 때까지 그대로의 상태를 유지하며 대기한다.

또, 래스터 변환회로(13)이 FIFO(16)에서 데이터를 리드할 때에 FIFO(16)의 기억영역에 데이터가 전혀 들어가지 않으면(엠프티 상태), 리드가 되지 않아 데이터가 FIFO(16)에 입력될 때까지 그대로의 상태로 대기 하지 않으면 안된다.

제2도는 이들 상술한 레지스터(11)과 연산기(12)의 상세한 종속 접속 구성을 도시한다. 사용자의 제1도의 CPU(1)에 접속되는 콘솔(도시하지 않음)을 거쳐서 사전에 소정의 좌표 매트릭스 데이터 a, b, c, d, Tx, Ty를 입력하고, 이들 데이터는 레지스터(11a)에 유지된다. 일시 레지스터(11b), 포인트 데이터(X, Y)용 레지스터(11c)도 레지스터(11a)와 함께 레지스터(11)에 포함되고, 도시한 라인 A, B에 의해 접속되어 있다. 이들 레지스터의 데이터는 다음에 기술하는 방법으로 모두 연산기(12) 내에서 접속된 승산기(12a), ALU(12b)에 의해 연산되어 FIFO(16)에 입력된다.

다음에 본 발명의 주요부를 구성하는 문자 도형 발생회로(4)의 동작에 대해서 제1도~제3도를 사용해서 상세하게 설명한다.

제3도는 데이터 변환회로(6)의 동작의 알고리즘을 나타낸 흐름도이다.

우선, CPU(1)는 문자 도형 발생회로(4)에 기동을 건다. 기동이 걸리지 않으면 그대로의 상태로 대기한다(스텝21).

데이터 변환회로(6)는 프로그램 ROM(5)의 내용을 PC(9)의 카운트 값에 의해서 지정되는 어드레스에서 순차 리드하고(스텝22), 리드한 명령을 디코드한다(스텝23).

디코드된 명령에 따라서 데이터 변환 실행 유닛(10)내의 레지스터군(11) 및 연산기(12)를 사용해서 벡터 형식의 문자 도형 데이터에 소정의 데이터 신장(폰트 ROM내의 데이터가 압축되어 있는 경우)을 가하고(스텝24), 신장후의 데이터에 확대나 축소 등의 좌표 변환을 가한다(스텝25).

다음에, 좌표 변환후의 데이터에 선 폭 등을 일치시키는 보정 처리를 가하고(스텝26), 보정 처리후의 데이터에 곡선 보간을 가한다(스텝27).

마지막으로 곡선 보간후의 데이터를 FIFO(16)으로 전송하고(스텝28), 스텝22부터의 처리를 반복한다.

이 스텝21부터 스텝26까지의 동작을 실행하는 것에 의해, 벡터 형식의 문자 도형 데이터에 데이터 신장, 좌표 변환, 보정, 곡선 보간 등의 변환을 가할 수가 있다.

다음에 실제로 「P」인 문자의 작성에 대해서 아웃 라인 폰트 데이터에 페인팅이 실행된 비트 맵 데이터를 발생시킬 때까지 필요한 처리 및 각 처리의 처리량에 대해서 제4도, 제5도를 사용해서 더욱 상세하게 설명한다.

페인팅은 좌측에서 우측을 향해서 비트 맵 데이터를 주사하고, 값1을 나타내는 데이터를 찾아서 홀수번째에서 짝수번째의 1 사이의 데이터 0을 1로 변환하는 작업이다.

또한, 곡선 보간후의 데이터에서 도트 형식의 문자 도형 데이터를 발생시키는 처리에 대해서는 전용 회로, 흐름도를 사용해서 다음에 다시 설명한다.

제4도는 압축된 아웃 라인 폰트 데이터에서 페인팅이 실행된 비트 맵 데이터를 발생시키는 흐름을 나타낸 모식도이다.

또, 제5도는 아웃 라인 폰트에 가하는 처리 전체에 대해서 각 처리가 차지하는 시간적인 비율을 나타낸 원그래프이다.

제4도에 있어서 (201)은 압축된 아웃 라인 폰트 데이터이고, 통신 폰트 ROM 용량 삭감을 위해 아웃 라인 폰트 데이터는 압축되어 기억되어 있다.

(202)는 압축된 아웃 라인 폰트 데이터에 신장 처리를 가한 후의 데이터이다.

(203)은 신장된 데이터에 확대나 축소 등의 좌표 변환을 가한 후의 데이터이다.

(204)는 좌표 변환된 데이터에 문자 도형의 선폭을 일치시키기 위한 보정 처리를 가한 후의 데이터이다.

(203)에 있어서는 선폭이 3비트인데 대해, (204)에 있어서는 선폭이 2비트로 되어 있다.

또, (205)는 보정 처리된 데이터에 매끄러움을 가하기 위한 곡선 보간 처리를 실시한 데이터이다.

또, (206)은 (205)의 데이터 사이에 직선을 발생시키고, 문자 도형의 윤곽선을 발생시키기 위한 윤곽선 발생처리를 실시한 데이터이다.

마지막으로 (207)은 (206)의 데이터에 페인팅 처리를 가한 후의 데이터로서, 이(207)의 데이터가 문자 도형의 비트 맵 데이터로 된다.

데이터(201)이 (207)에 도달할 때까지의 처리에 대해서 설명한다.

우선, (201)의 아웃 라인 폰트 데이터는 통상 상대 좌표값으로 압축되어 있으므로, 데이터 신장 처리는 상대 좌표값을 순차 가산해 가며 신장한다.

이 때문에, 데이터 신장을 위해서 필요한 처리는 X, Y좌표의 증분값의 가산이다.

또, 좌표 변화는 좌표값(X, Y)를 기본으로 $XA=aX+bY+Tx$, $YA=cX+dY+Ty$ (단, XA, YA는 좌표 변환후의 X, Y 좌표, a, b, c, d, Tx, Ty는 사전에 레지스터(11a)에 입력되어 있던 좌표 매트릭스 데이터이고, a는 1행 1열 요소, b는 1행 2열 요소, c는 2행 1열 요소, d는 2행 2열 요소, Tx는 1행 3열 요소, Ty는 2행 3열 요소이다)로 되는 선형 변환식에 의해, 데이터의 확대 축소 등을 실행하는 처리이며, 상기 처리를 실시한다. 이 2개의 식의 우변의 계산 도중의 데이터는 일시 레지스터(11b)에 들어가고, 최종 결과, 즉 좌변의 결과는 레지스터(11c)에 들어간다. 레지스터(11c)는 4점의 변환후의 좌표값을 갖는다.

상기 식에 나타낸 바와 같이, 좌표 변환에 필요한 처리는 X, Y 좌표값에 대한 계수의 승산 및 가산이고, 승산기와 ALU(12a), (12b)에 의해 실행된다.

다음에 보정 처리는 지정된 좌표를 안쪽으로 이동시키는 것에 의해, 문자 도형의 선풍을 일치시키는 처리이다.

이동시키는 좌표의 수는 1문자 도형당 수소이므로, 처리량은 극히 적지만 처리 자체는 이동에 관한 각종의 판정등 때문에 복잡하다.

다음에 곡선 보간은 지정된 영역내에 새롭게 수정의 문자 발생을 위한 포인트를 발생시키고, 발생시키는 문자 매끄러움을 가하는 처리이다.

지정되는 영역은 1문자 도형당 소량이므로, 처리수 자체는 적지만 처리 자체는 추가 포인트의 판단 등 때문에 복잡하다.

다음에 윤곽선 발생처리는 좌표 사이에 직선을 발생시켜서 문자 도형의 윤곽선을 발생시킨다.

좌표 사이에 직선을 발생시키는 처리는 좌표 사이를 점열로 연결할 수 있는 것이므로, 발생시키는 점열의 수는 방대하다.

마지막으로 페인팅 처리는 문자 도형의 윤곽선의 내부를 점(도트)으로 메우는 것이며, 발생시키는 점(도트)의 수는 방대하다.

따라서, 윤곽선의 발생 및 페인팅에 필요한 처리는 극히 다량이다.

상술한 바와 같이, 데이터 신장, 좌표 변환, 보정, 곡선 보간은 비교적 처리량은 적지만 처리 자체는 복잡하다.

이것에 대해서, 윤곽선 발생, 페인팅 등은 처리 자체는 단순하지만 처리량은 방대하다.

제5도에 아웃 라인 폰트 전개 시간 전체가 차지하는 각 처리 마다의 대략적인 처리 시간의 구성을 도시한다.

제5도에서 명확한 바와 같이, 다소의 오차는 있지만, 윤곽선 발생 및 페인팅 처리가 전체의 65%를 차지하고 있다.

데이터 신장, 좌표 변환, 보정, 곡선 보간 등의 비교적 처리량이 적고 복잡한 처리는 소정의 프로그램에 따라서 1스텝마다 동작하는 회로를 사용해서 저속으로 실행하고, 윤곽선 발생, 페인팅 처리 등과 같이 처리량이 매우 많으므로 고속의 실행이 요구되는 처리는 전용의 회로를 사용해서 고속으로 실행하는 것이 바람직하다.

따라서, 윤곽선 발생, 페인팅 처리만 전용의 회로에 의해 고속으로 실행하고, 그 이외의 처리를 소정의 프로그램에 따라서 1스텝마다 동작하는 회로에 의해 저속으로 실행하는 것에 의해서 처리 속도 및 기능을 떨어뜨리지 않고 회로 규모를 억제할 수가 있다.

또, 소정의 프로그램에 따라 1스텝마다 동작하는 회로는 연산기, 디코더, 레지스터, 프로그램 ROM 등을 공유할 수 있으므로, 각 처리마다 전용의 연산기, 레지스터 등의 회로를 갖는 경우보다도 회로 규모가 극히 작아도 좋다.

또, 소정의 프로그램에 따라서 1스텝마다 동작하는 회로는 전용의 회로에 비해서 실행속도가 저속이기는 하지만, 그 회로에서 실행하는 처리는 비교적 처리량이 적기 때문에 저속인 것은 그다지 문제로 되지 않는다.

다음에, 만약 폰트 ROM(2)에 저장된 벡터 형식의 문자 도형 데이터가 프로그램 ROM(5)에 저장되어 있는 프로그램에 대응하고 있지 않은 폰트 데이터에 따르고 있으면 정확하게 동작할 수 없는 경우가 있다. 다음에, 여러가지 포맷의 폰트 데이터로의 대응 방법을 제6도~제8도를 사용해서 상세하게 설명한다.

우선, 제1의 방법에 대해서 제6도를 사용해서 설명한다.

본 실시예는 데이터 변환회로(6), 래스터 변환회로(13), FIFO(16), 워크 메모리(17), 리라이트 가능한 프로그램 ROM(18) 및 ROM 라이터(31)를 갖고 구성된다.

데이터 변환회로(6), 래스터 변환회로(13), FIFO(16), 워크 메모리(17)는 제1도에서 설명한 구성요소와 동일한 것을 사용해서 실현할 수 있기 때문에, 여기에서는 상세한 설명은 생략한다.

프로그램 ROM(18)은 프로그램을 기억하는 수단으로서 반도체 IC인 소위 PROM(프로그램 롬) 등으로 구성된다.

ROM 라이터(31)는 프로그램 ROM(18)에 데이터를 라이트하기 위한 수단으로서, CPU, 각종 TTL, 데이터버스, 인터페이스 커넥터 등의 전자 디바이스로 구성된다.

또, 프로그램 ROM(18)에는 이미 폰트 데이터의 포맷에 대응하고 있지 않은 프로그램이 저장되어 있는 것으로 한다.

이 상태에서는 정확하게 동작하지 않으므로, ROM 라이터(31)를 사용하여 폰트 포맷에 대응한 프로그램을 다시 라이트한다.

여기에서 ROM 라이터(31)와 제1도의 문자 도형 발생 장치는 별개의 장치로서, ROM 라이터(31)에 의해 프로그램 ROM(18)에 데이터를 전기적으로 라이트한다.

폰트 포맷에 대응한 프로그램을 라이트한 후에 제3도의 스텝21부터 스텝28까지의 동작을 실행하는 것에 의해, 벡터 형식의 문자 도형 데이터에 데이터 신장, 좌표 변환, 보정, 곡선 보간 등의 변환을 가하는 것이 가능하게 된다.

일반적으로 PROM은 리라이트 가능하고, 폰트 변경에 대응할 수 있다.

또, 제조자측이 출하 시점에서 고객이 요구하는 폰트에 따른 처리 프로그램을 한번만 라이트하는 것만으로 좋은 경우에는 한번만 라이트 가능한 ROM을 사용하면 좋다.

다음에 제2의 방법에 대해서 제7도, 제8도를 사용해서 상세하게 설명한다.

제7도는 워드 프로세서, 퍼스널 컴퓨터, 프린터 등의 정보처리 장치에 있어서 제2의 방법을 실행하는 블록도의 1예이고, 제8도는 제7의 문자 도형 발생 장치의 동작의 알고리즘을 나타낸 흐름도이다.

본 실시예에서는 CPU(1), 폰트 ROM(2) 및 출력 엔진(3), 문자 도형 발생회로(41)를 갖고 구성된다. 여기에서, CPU(1), 폰트 ROM(2) 및 출력엔지(3)은 제1도에서 설명한 실시예와 동일한 것을 사용할 수 있으므로 상세한 설명은 생략한다.

또, 문자 도형 발생 회로(41)은 프로그램 RAM(42), 데이터 변환 회로(6), FIFO(16), 래스터 변환 회로(13)를 갖고 구성된다.

데이터 변환 회로(6), FIFO(16), 래스터 변환 회로(13)는 제1도에서 설명한 실시예와 동일한 것을 사용할 수 있기 때문에, 상세한 설명은 생략한다.

프로그램 RAM(42)은 벡터 형식의 문자 도형 데이터에 데이터 신장, 좌표 변환, 보정, 곡선 보간 등의 처리를 실행하는 프로그램을 저장하는 수단으로서, 반도체 메모리 등의 전자 디바이스로 구성된다.

다음에, 제7도의 문자 도형 발생 장치(2)를 사용해서 문자 도형을 출력하는 순서에 대해서 제8도를 사용해서 상세하게 설명한다.

우선, CPU(1)은 프로그램 RAM(42)에 프로그램을 라이트 하고(스텝51), 문자 도형 발생 회로(41)에 기동을 건다(스텝52).

기동이 걸리지 않으면 그대로의 상태로 대기한다.

데이터 변환 회로(6)은 프로그램 RAM(42)의 내용을 순차 리드하고(스텝53), 리드한 명령을 디코드한다(스텝54).

디코드된 명령에 따라서 데이터 변환 실행 유닛(10)내의 레지스터군(11) 및 연산기(12)를 사용하여 벡터 형식의 문자 도형 데이터에 데이터 신장(폰트 ROM(2)내에 데이터가 압축되어 있는 경우)를 가하고(스텝55), 신장후의 데이터에 확대나 축소등의 좌표 변환을 가한다(스텝56).

다음에, 좌표 변환후의 데이터에 선폭 등을 일치시키는 보정 처리를 실시하고(스텝57), 보정 처리후의 데이터에 곡선 보간의 처리를 실시한다(스텝58).

마지막으로, 곡선 보간후의 데이터를 FIFO(16)로 전송하고(스텝59), 스텝53부터의 처리를 반복한다.

이 스텝51부터 스텝59까지의 동작을 실행하는 것에 의해, 벡터 형식의 문자 도형 데이터에 데이터 신장, 좌표 변환, 보정, 곡선 보간 등의 처리를 실시할 수 있다.

이상 기술한 제1 또는 제2의 방법을 사용하는 것에 의해, 벡터 형식의 문자 도형 데이터가 각종의 포맷을 갖고 있더라도 도트 형식의 문자 도형 데이터를 간단하게 발생시킬 수가 있다.

이미, 실제로 「P」 인 문자의 작성에 대해서 아웃 라인 폰트 데이터에서 페인팅이 실행된 비트 맵 데이터를 발생시킬 때까지 필요한 처리 및 각 처리의 처리량에 대해서 제4도, 제5도로 사용하여 설명하였지만, 마지막으로 곡선 보간후의 데이터에서 도트 형식의 문자 도형 데이터를 발생시키는 순서에 대해서 제9도에 도시한 전용회로, 제10도에 도시한 처리 흐름도를 사용해서 다시 설명한다.

제9도는 제1도에 있어서의 래스터 변환회로(13)의 상세 설명도이고, 제10도는 래스터 변환회로(13)의 동작의 알고리즘을 나타낸 흐름도이다.

제9도에 도시한 래스터 변환회로(13)은 콘트롤부(14) 및 래스터 변환 실행 유닛(15)를 갖고 구성된다.

또, 콘트롤부(14)는 윤곽선 발생 제어부(61) 및 페인팅 제어부(63)를 갖고 구성된다.

여기에서, 윤곽선 발생 제어부(61)는 좌표점 사이에 직선을 발생시켜서 문자 도형의 윤곽선을 워크 메모리(17)상에 발생시키는 처리를 제어하는 수단으로서, 각종 TTL 등의 전자 디바이스로 구성된다.

여기에서, 페인팅 제어부(63)는 문자 도형의 윤곽선의 내부를 페인팅하는 처리를 제어하는 수단으로서, 각종 TTL 등의 전자 디바이스로 구성된다.

또, 래스터 변환 실행 유닛(15)은 윤곽선 발생 실행 유닛(62) 및 페인팅 실행 유닛(64)을 갖고 구성된다.

여기에서, 윤곽선 발생 실행 유닛(62)은 윤곽선 발생에 필요한 연산을 실제로 실행하는 수단으로서, 각종 TTL 등의 전자 디바이스로 구성된다.

여기에서, 페인팅 실행 유닛(64)은 페인팅에 필요한 연산을 실제로 실행하는 수단으로서, 각종 TTL 등의 전자 디바이스로 구성된다.

다음에, 래스터 변환회로(13)를 사용하여 곡선 보간후의 데이터에서 도트 형식의 문자 도형 데이터를 워크메모리(17)상에 발생시킬 때까지의 순서에 대해서 제10도를 사용해서 설명한다.

우선, 윤곽선 발생 제어부(61)는 제1도에 도시한 FIFO(16)의 값을 리드하고, 좌표 데이터를 받아들인다(스텝71).

상기 좌표 데이터 사이에 라인을 발생하고, 워크 메모리(17)상에 묘화한다(스텝72).

1문자 도형에 대해서 윤곽선의 발생이 종료하였다면, 페인팅 제어부(63)에 기동을 걸고, 종료하지 않았다면 스텝71부터의 처리를 반복하여 윤곽선의 발생처리를 계속한다(스텝73, 74).

기동이 걸린 페인팅 제어부(63)는 워크 메모리(17)상의 데이터를 리드하고(스텝75), 윤곽선 내부를 페인팅한다(스텝76).

1문자 도형본의 윤곽선의 내부를 모두 페인팅하였으면 페인팅이 종료한 것을 CPU(1)에 알리고, 페인팅이 완전히 종료하지 않았으면 스텝75부터의 동작을 반복하여 페인팅을 계속한다(스텝77, 78).

이 스텝71부터 스텝78까지의 동작을 실행하는 것에 의해, 워크 메모리(17)상에 문자 도형의 윤곽선을 발생하고, 내부를 페인팅할 수가 있다.

윤곽선 발생, 페인팅의 상세한 내용에 대해서는 일본국 특허 공개 공고 소화59-71093호에 기재되어 있다.

이상, 상술한 바와 같은 회로구성을 갖고, 데이터 신장, 좌표변환, 보정, 곡선 보간 등의 비교적 처리량이 적은 처리는 데이터 변환 회로(6)와 같은 프로그램을 사용하여 1스텝마다 처리가 실행되는 회로를 사용해서 저속도로 처리한다.

이것에 의해서, 회로 규모를 억제하는 것이 가능하게 된다.

이것에 대해서 윤곽선 발생, 페인팅 등의 처리량이 많은 처리에 대해서는 래스터 변환회로(13)와 같은 전용회로를 사용하여 고속으로 실행한다.

즉, 처리량이 적은 처리는 프로그램을 사용하여 1스텝마다 처리가 실행되는 회로로 처리를 실행하여 회로 규모를 억제하고, 처리량이 많은 처리만 전용의 회로에 의해 고속으로 실행하는 것이다.

이것에 의해서, 회로 규모를 크게 하는 일없이 고속이며 또한 효율적으로 도트 형식의 문자 도형 데이터를 발생시킬 수가 있다.

또, 본 실시예에서는 데이터 변환회로(6) 내에 데이터 신장, 좌표 변환, 보정, 곡선 보간의 4가지 처리를 조립해서 설명을 하였지만, 상기 4가지의 처리중 필요한 것만으로 조립하는 구성 또는 이들 이외의 처리를 조립하는 구성으로 하여도 좋다.

제11도는 또 다른 실시예를 도시한 것이다.

사전에 프로그램을 하드 디스크 HD 또는 플로피 디스크 FD 등의 디스크 장치(82)에 저장해 두고, HD, FD의 프로그램을 문자 발생 LSI의 프로그램 RAM(42)에 라이트한다. 그 이외의 회로 블록의 구성은 제1도와 동일하여도 좋다.

제12도는 다른 실시예를 도시한 것으로, 문자 도형 발생 회로(4)는 도시한 바와 같이 프로그램 ROM(5)에 접속되어 소정의 전류를 상기 ROM 내의 선택된 어드레스의 선택된 비트에 흐르게 하기 위해서, 검증하기 위한 PROM 라이터 회로(93)를 갖고 있다. 라이터 회로는 ROM 라이터(96)에 접속되어 어드레스와 비트를 ROM라이터에서 지시받아 상술한 ROM(5)으로의 라이트를 실시한다. 따라서, 이 실시예에 의하면 불휘발성 메모리로의 라이트를 문자 도형 발생 회로의 사용자가 실시할 수 있다.

제13도는 버스를 거쳐서 문자 도형 발생 회로(41)상의 프로그램 RAM(42)에 접속된 터미널 디바이스(101)에 의해, A, B의 양폰트 ROM 중에서 선택/입력되는 폰트 타입에 따라서 A폰트 데이터를 갖는 ROM(2) 또는 B폰트 데이터를 갖는 ROM(2)의 어느 것인가 하나가 선택되고, 어느 것인가 한쪽의 폰트 데이터에 대응하는 프로그램이 디스크 장치(82)중에서 리드되어 프로그램 RAM(42)에 라이트되는 문자 도형 시스템(102)을 도시한다.

이상 기술한 실시예에 있어서, 동일 블록 번호를 갖는 블록의 설명은 이미 기술한 일련의 실시예에 있어서의 기능 설명과 마찬가지로 생략되어 있다.

제14도는 제13도에 도시한 시스템에 또 하나의 LSI(4)를 버스(18)에 접속한 다른 실시예를 도시한 것이다. 버스(18)를 거쳐서 2개의 LSI(4)에 접속되는 프로그램 유지 유닛(112)은 프로그램 RAM(42)과 (42)에 기억해야 할 여러개의 명령군 세트 A, B, C, ...를 갖고 있고, CPU(1)에 접속되는 터미널 디바이스(도시하지 않음)를 거쳐서 입력되는 지정에 따라서 바라는 명령군 세트가 각각의 LSI(4)의 RAM(42)와 (42)에 로드된다. 이 실시예에서는, 사용자가 바라는 여러개의 폰트의 조합을 시스템상으로 구축할 수가 있다.

제15도는 제7도에 도시한 버스(18)에 접속된 폰트 ROM(2) 대신에 2개의 폰트 타입에 대응하는 카세트화된 ROM의 형태를 취하는 2개의 폰트 카세트 ROM(132, 134)과 한쪽의 폰트에 따른 프로그램을 프로그램 RAM(42)으로 보내는 플로피 디스크 장치(133)이 접속된 다른 실시예를 나타내었다.

다른 쪽의 폰트에 대응하는 프로그램은 다른쪽의 폰트 카세트 ROM 중에 폰트 데이터와 함께 기억되어 있다. 폰트 ROM 카세트(132)형의 카세트를 여러개 또는 폰트 ROM 카세트(134)형의 카세트를 여러개 구비한 구성의 시스템을 구축하는 것도 가능하다.

(57) 청구의 범위

청구항 1

윤곽 데이터를 비트 맵 데이터로 변환하는 회로 블록으로서, 폰트 데이터를 소정의 파라미터(a, b, ...Ty)에 따라 변환하기 위한 여러개의 명령을 기억하기 위한 메모리(5), 상기 메모리에 접속되고, 폰트 데이터에 상기 메모리에 기억된 여러개의 명령에 의해서 지시되는 변환을 실시하기 위한 제1데이터 변환 회로 수단(6), 상기 레지스터와 상기 연산기에 접속되고, 상기 변환을 실시한 데이터를 순차적으로 받아들여서 유지하고 그 데이터를 받아들인 순서대로 출력하는 선입 선출 수단(16), 상기 선입 선출 수단(16)에 접속되고, 상기 선입 선출 수단에서 순서대로 출력되는 상기 데이터를 비트 맵 데이터로 변환하는 제2데이터 변환 회로 수단(13), 상기 제2데이터 변환 회로 수단에 접속되고, 상기 비트 맵 데이터를 유지하는 비트 맵 메모리(17)를 포함하고, 상기 제1데이터 변환 회로 수단은 상기 소정의 파라미터를 기억하는 레지스터(11a)와 상기 폰트 데이터를 연산하기 위한 연산기(12)를 갖고, 상기 제1데이터 변환 회로 수단은

상기 선입 선출 수단이 유지 데이터로 풀 상태인 경우는 데이터 변환하는 것을 중단하고 엠프티영역이 생기는 것을 대기하며, 상기 제2데이터 변환 회로 수단은 상기 선입 선출 수단 내에 데이터가 엠프티 상태이면 데이터가 그 내부에 유지될 때까지 대기하는 문자 도형 발생 회로.

청구항 2

특허청구의 범위 제1항에 있어서, 상기 제2데이터 변환 회로 수단은 상기 데이터를 래스터화하기 위한 래스터 변환 실행 유닛(15)과 상기 래스터 변환 실행 유닛을 제어하는 컨트롤러(14)를 갖고, 상기 비트 맵 메모리 중의 변환된 비트 맵 데이터의 윤곽선 내부 부분의 페인트를 실행하는 문자 도형 발생 회로.

청구항 3

특허청구의 범위 제2항에 있어서, 상기 제2데이터 변환 회로 수단은 상기 윤곽선 내부 부분의 페인트 종료 후, 페인트 종료 신호를 발생하는 종료 신호 발생 수단을 갖는 문자 도형 발생 회로.

청구항 4

특허청구의 범위 제1항에 있어서, 상기 회로 블록은 하나의 대규모 집적 회로내에 구성되고, 상기 메모리는 프로그램 가능한 리드 온니 메모리인 문자 도형 발생 회로.

청구항 5

특허청구의 범위 제1항에 있어서, 제1데이터 변환 회로 수단은 상기 신장된 데이터를 좌표 변환하는 수단과 상기 신장된 데이터를 위한 소정의 수점의 포인트 데이터를 발생하기 위한 곡선 보간 수단을 갖는 문자 도형 발생 회로.

청구항 6

특허청구의 범위 제1항에 있어서, 상기 회로 블록은 하나의 대규모 집적 회로내에 구성되고, 상기 메모리는 랜덤 액세스 메모리인 문자 도형 발생 회로.

청구항 7

윤곽 데이터를 비트 맵 데이터로 변환하는 시스템으로서, CPU(1), 상기 CPU(1)에 접속되고, 폰트 데이터를 갖는 폰트 ROM(2), 상기 CPU 및 상기 ROM에 접속되고, 아웃 라인 폰트 데이터를 입력하고 비트 맵 데이터로 변환하는 회로 블록(4)을 포함하고, 상기 회로 블록은 상기 폰트 ROM에 접속되고, 폰트 데이터를 소정의 파라미터(a, b, ..., Ty)에 따라 변환하기 위한 여러개의 명령을 기억하기 위한 랜덤 액세스 메모리(42), 상기 메모리에 접속되고, 폰트 데이터에 상기 메모리에 기억된 여러개의 명령에 의해서 지시되는 변환을 실시하기 위한 제1데이터 변환 회로 수단(6), 상기 레지스터와 상기 연산기에 접속되고, 상기 변환을 실시한 데이터를 순차적으로 받아들여서 유지하고 그 데이터를 받아들인 순서대로 출력하는 선입 선출 수단(16), 상기 선입 선출 수단에 접속되고, 상기 선입 선출 수단에서 순서대로 출력되는 상기 데이터를 비트 맵 데이터로 변환하는 제2데이터 변환 회로 수단(13), 상기 제2데이터 변환 회로 수단에 접속되고, 상기 비트 맵 데이터를 유지하는 비트 맵 메모리(17)를 포함하며, 상기 제1데이터 변환 회로 수단은 상기 소정의 파라미터를 기억하는 레지스터(11a)와 상기 폰트 데이터를 연산하기 위한 연산기(12)를 갖고, 상기 제1데이터 변환 회로 수단은 상기 선입 선출 수단이 유지 데이터로 풀 상태인 경우는 데이터 변환하는 것을 중지하고 엠프티 영역이 생기는 것을 대기하며, 상기 제2데이터 변환 회로 수단은 상기 선입 선출 수단의 중에 데이터가 엠프티 상태이면 데이터가 그 속에 유지될 때까지 대기하는 문자 도형 변환 시스템.

청구항 8

특허청구의 범위 제7항에 있어서, 상기 제2데이터 변환 회로 수단은 상기 데이터를 래스터화하기 위한 래스터 변환 실행 유닛(15)과 상기 래스터 변환 실행 유닛을 제어하는 컨트롤러(14)를 갖고, 상기 비트 맵 메모리 중의 변환된 비트 맵 데이터의 윤곽선 내부 부분의 페인트를 실행하는 문자 도형 변환 시스템.

청구항 9

특허청구의 범위 제8항에 있어서, 상기 제2데이터 변환 회로 수단은 상기 윤곽선 내부 부분의 페인트 종료 후, 페인트 종료 신호를 발생하는 종료 신호 발생 수단을 갖는 문자 도형 변환 시스템.

청구항 10

특허청구의 범위 제7항에 있어서, 제1데이터 변환 회로 수단은 상기 신장된 데이터를 좌표 변환하는 수단과 상기 신장된 데이터를 위한 소정의 수점의 포인트 데이터를 발생하기 위한 곡선 보간 수단을 갖는 문자 도형 변환 시스템.

청구항 11

윤곽 데이터를 비트 맵 데이터로 변환하는 시스템으로서, CPU(1), 상기 CPU에 접속되고, 폰트 데이터를 갖는 폰트 ROM(2), 상기 CPU에 접속되고, 폰트 데이터를 소정의 파라미터(a, b, ..., Ty)에 따라 신장하기 위한 여러개의 명령을 기억하기 위한 디스크 장치(82), 상기 CPU, 상기 디스크 장치 및 상기 ROM에 접속되고, 아웃 라인 폰트 데이터를 입력하고 비트 맵 데이터로 변환하는 회로 블록(4)을 포함하고, 상기 회로 블록은 상기 디스크 장치 및 상기 폰트 ROM에 접속되고, 폰트 데이터를 상기 소정의 파라미터(a, b, ..., Ty)에 따라 변환하기 위한 상기 여러개의 명령을 기억하기 위한 랜덤 액세스 메모리(42), 상기 메모리에 접속되고, 폰트 데이터에 상기 메모리에 기억된 여러개의 명령에 의해서 지시되는 변환을 실시하기 위한 제1데이터 변환 회로 수단(6), 상기 레지스터와 상기 연산기에 접속되고, 상기 변환을 실시한 데이터를 순차적으로 받아들여서 유지하고 그 데이터를 받아들인 순서대로 출력하는 선입 선출 수단(16), 상기 선입 선출 수단에 접속되고, 상기 선입 선출 수단에서 순서대로 출력되는 상기 데이터를 비트 맵 데이

타로 변환하는 제2데이터 변환 회로수단(13), 상기 제2데이터 변환 회로 수단에 접속되고, 상기 비트 맵 데이터를 유지하는 비트 맵 메모리(17)를 포함하며, 상기 제1데이터 변환 회로 수단은 상기 소정의 파라미터를 기억하는 레지스터(11a)와 상기 폰트 데이터를 연산하기 위한 연산기(12)를 갖고, 상기 제1데이터 변환 회로 수단은 상기 선입 선출 수단이 유지 데이터로 풀 상태인 경우는 데이터 변환하는 것을 중지하고 앰프티 영역이 생기는 것을 대기하며, 상기 제2데이터 변환 회로 수단은 상기 선입 선출 수단의 중에 데이터가 앰프티 상태이면 데이터가 그 속에 유지될 때까지 대기하는 문자 도형 변환 시스템.

청구항 12

특허청구의 범위 제11항에 있어서, 상기 제2데이터 변환 회로 수단은 상기 데이터를 래스터화하기 위한 래스터 변환 실행 유닛(15)과 상기 래스터 변환 실행 유닛을 제어하는 컨트롤러(14)를 갖고, 상기 비트 맵 메모리 중의 변환된 비트 맵 데이터의 윤곽선 내부 부분의 페인트를 실행하는 문자 도형 변환 시스템.

청구항 13

특허청구의 범위 제12항에 있어서, 상기 제2데이터 변환 회로 수단은 상기 윤곽선 내부 부분의 페인트 종료후, 페인트 종료 신호를 상기 CPU를 향해서 발생하는 종료 신호 발생 수단을 갖는 문자 도형 변환 시스템.

청구항 14

특허청구의 범위 제11항에 있어서, 또 상기 CPU에 접속된 터미널 디바이스(101), 상기 폰트ROM(2)가 갖는 폰트 데이터와 다른 폰트 데이터를 갖는 별개의 폰트 ROM(2')를 포함하고, 상기 디스크 장치는 상기 다른 폰트 데이터를 별개의 소정의 파라미터(a, b, ..., Ty)에 따라 신장하기 위한 여러개의 별개의 명령을 기억하고, 상기 랜덤 액세스 메모리에는 상기 터미널 디바이스에 입력되는 지시에 따라서 상기 2종류의 여러개의 명령중 한쪽이 기억되는 문자 도형 변환 시스템.

청구항 15

특허청구의 범위 제11항에 있어서, 디스크 장치는 플로피 디스크 장치(133)이고, 상기 폰트 ROM은 카세트의 형식을 갖고 있고, 상기 카세트와 다른 별개의 폰트 데이터를 갖는 폰트 카세트 ROM(134)은 상기 별개의 폰트 데이터를 별개의 소정의 파라미터(a, b, ..., Ty)에 따라 신장하기 위한 별개의 여러개의 명령을 기억하는 문자 도형 변환 시스템.

청구항 16

특허청구의 범위 제15항에 있어서, 또 상기 CPU, 상기 디스크 장치 및 상기 ROM에 접속되고, 상기 비트 맵 데이터를 출력하기 위한 출력 엔진(3)을 포함하는 문자 도형 변환 시스템.

청구항 17

벡터 형식으로 표시된 윤곽 정보에서 도트 형식의 문자 도형 데이터를 발생하는 문자 도형 발생 회로에 있어서, 각종의 명령을 기억하는 제1의 기억영역, 상기 제1의 기억영역에 기억되어 있는 명령을 순차적으로 해석하고, 윤곽정보에 목적의 문자를 작성하기 위한 변환을 가하는 제1의 논리 유닛, 상기 변환이 가해진 윤곽 정보에 또 도트 형식의 데이터로의 변환을 가하는 제2의 논리 유닛 및 도트 형식의 문자 도형 데이터를 일시적으로 기억하는 제2의 기억영역을 하나의 LSI 칩상에 배치한 문자 도형 발생 회로.

청구항 18

특허청구의 범위 제17항에 있어서, 제1의 기억영역에 벡터형식으로 표시된 문자 도형의 윤곽정보에서 도트 형식의 문자 도형 데이터를 발생시키기 위해 필요한 처리의 일부를 표시하는 명령을 사전에 기억하고, 제1의 논리 유닛에는 제1의 기억영역에 기억된 명령을 해석하여 윤곽정보에 목적의 문자를 작성하기 위한 변환을 가해서 변환후의 윤곽정보를 제2의 논리 유닛으로 전송하는 기능을 구비하고, 제2의 논리 유닛에는 전송된 윤곽정보에 또 도트 형식의 데이터로서 변환을 가해서 도트 형식의 문자 도형 데이터를 제2의 기억영역에 발생시키는 기능을 구비한 문자 도형 발생 회로.

청구항 19

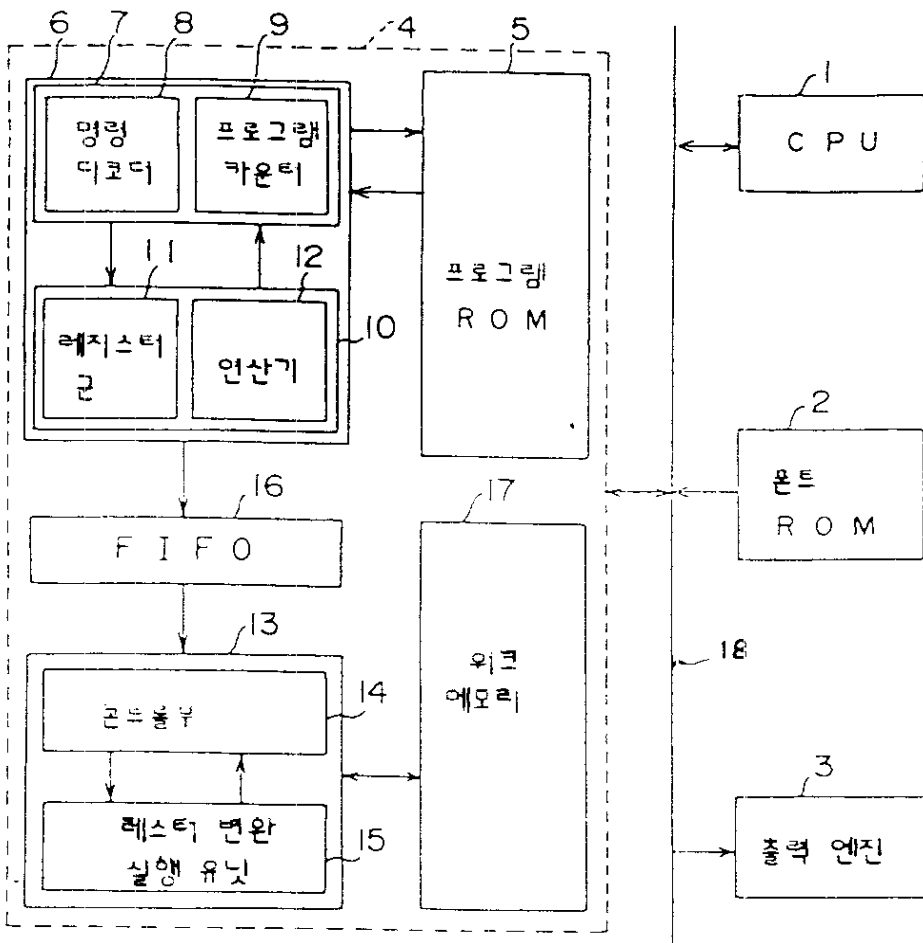
특허청구의 범위 제17항에 있어서, 제1의 기억 영역으로서 라이트 가능한 기억수단을 사용한 문자 도형 발생 회로.

청구항 20

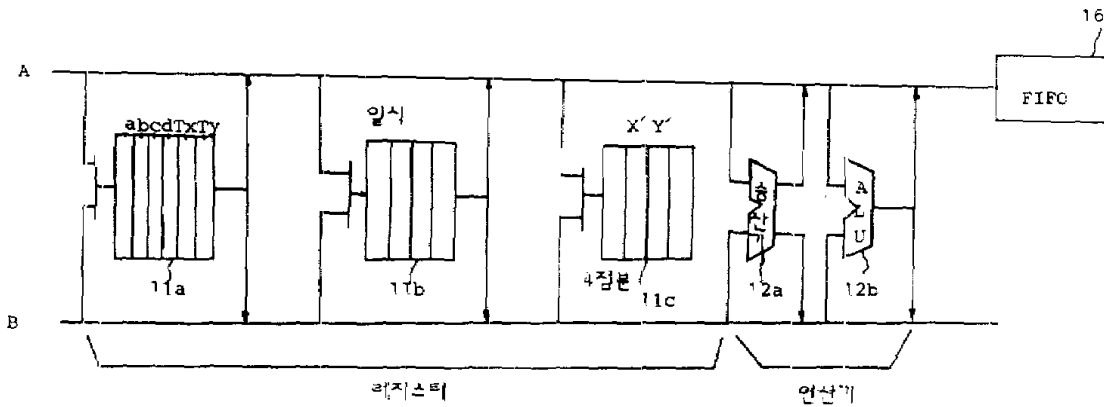
특허청구의 범위 제17항에 있어서, 제1의 기억영역으로서 RAM을 사용한 문자 도형 발생 회로.

도면

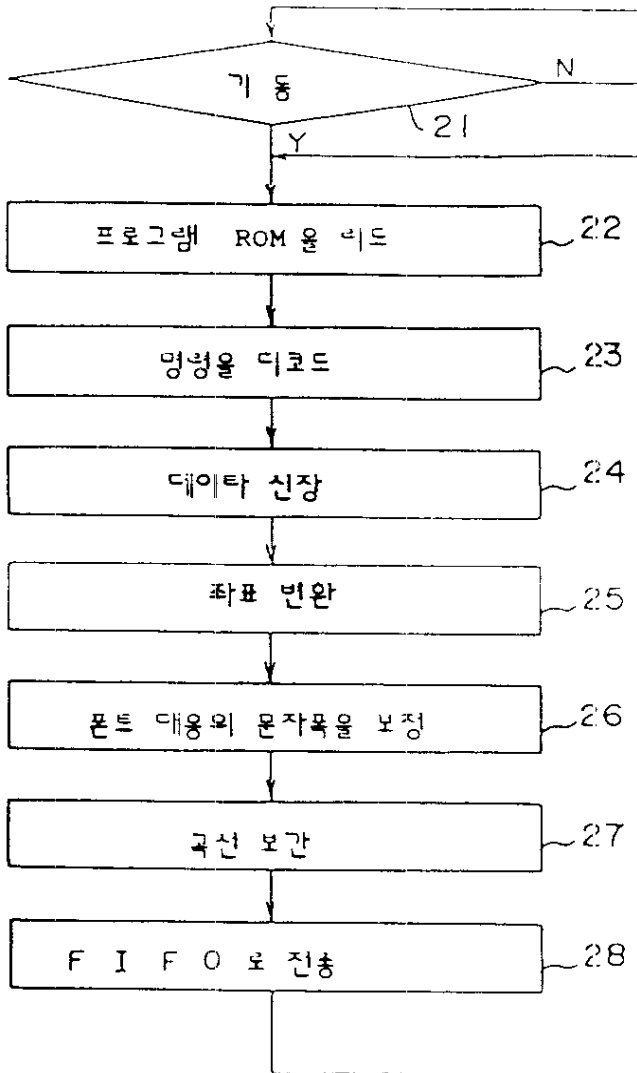
도면1



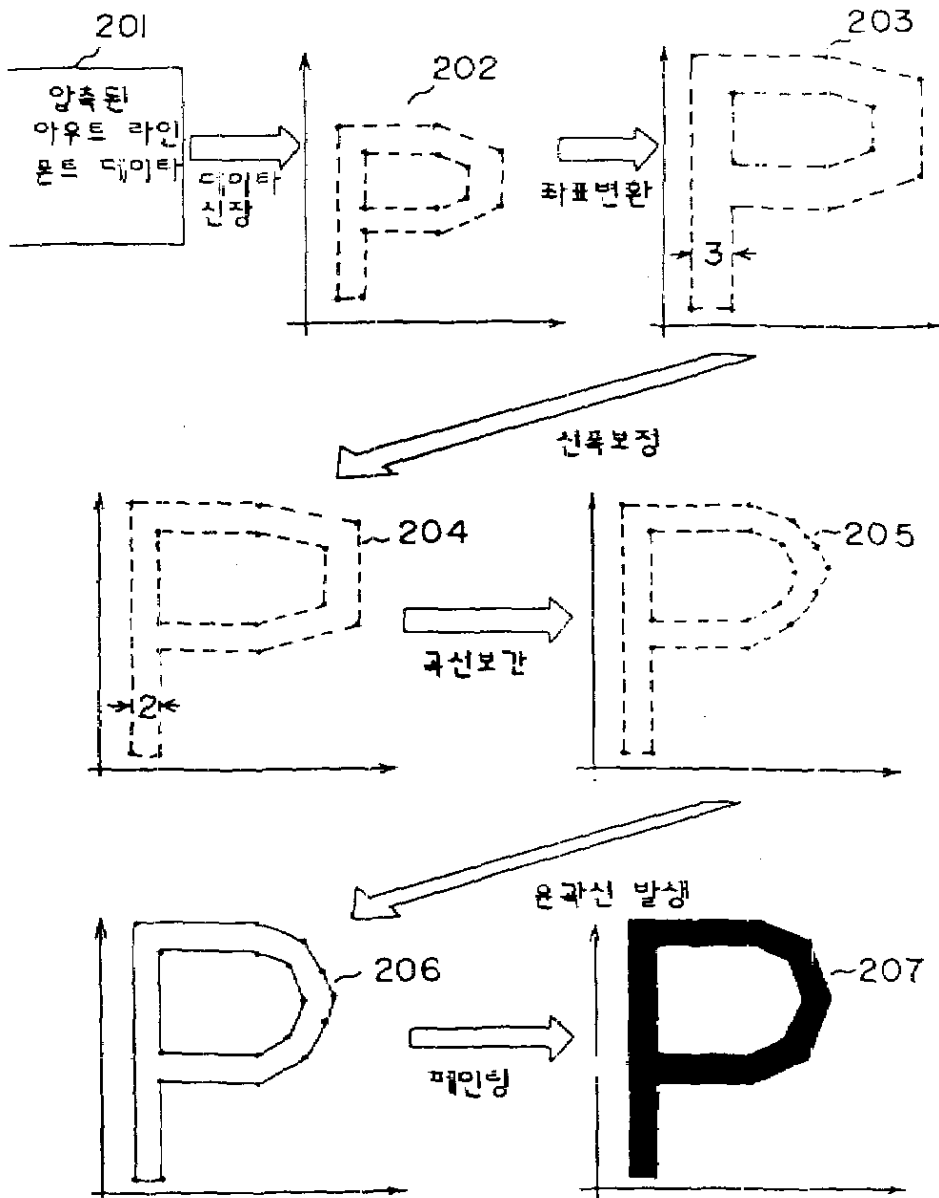
도면2



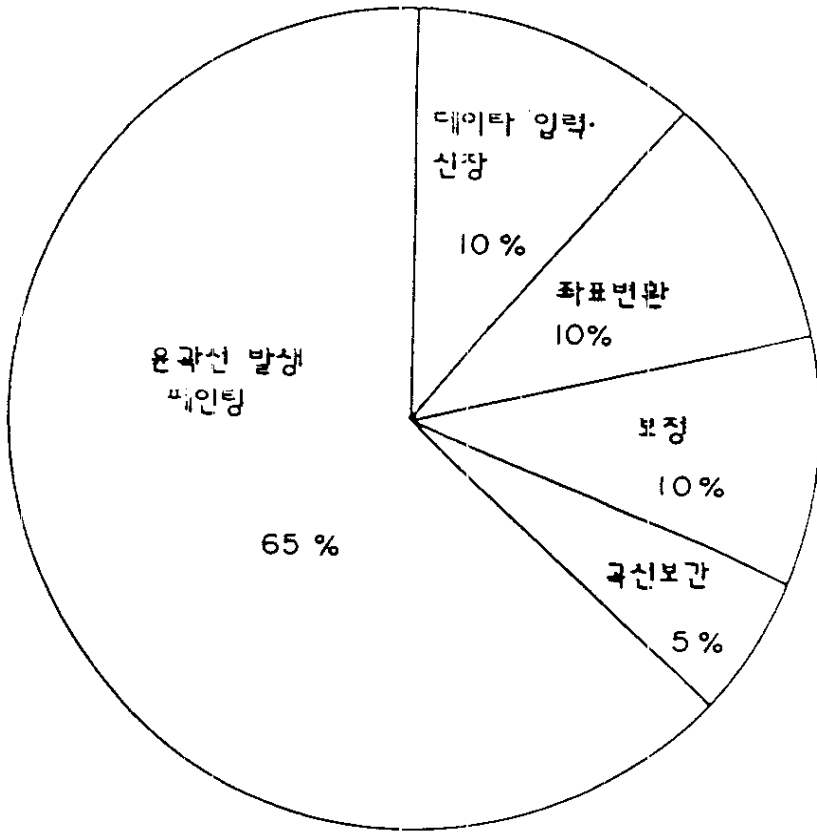
도면3



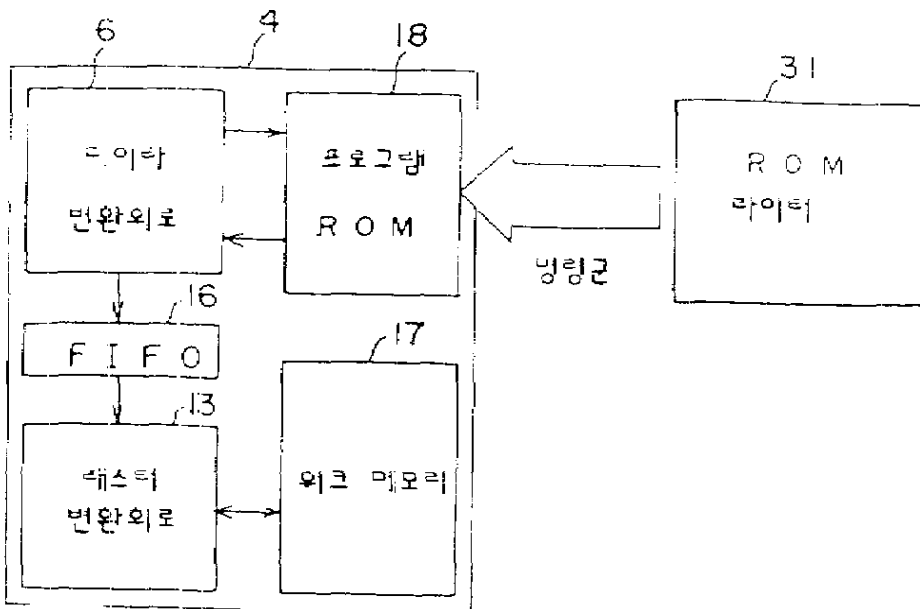
도면4



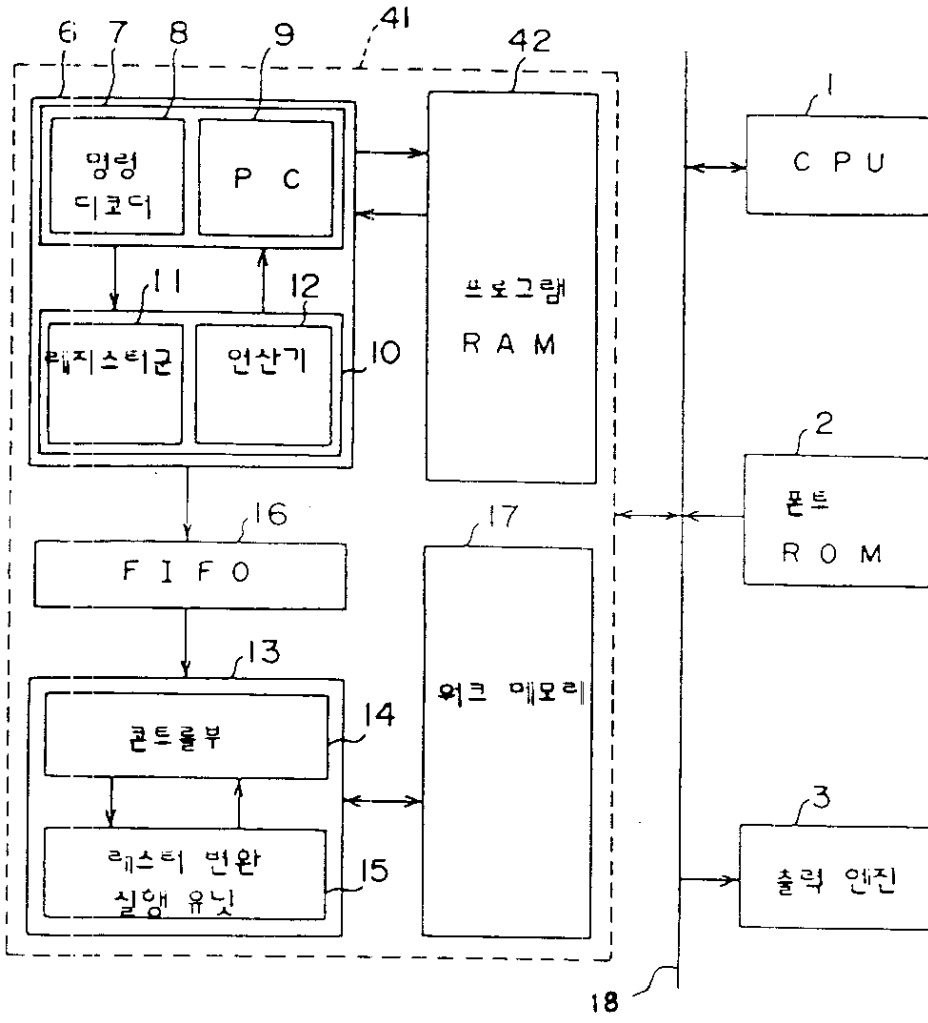
도면5



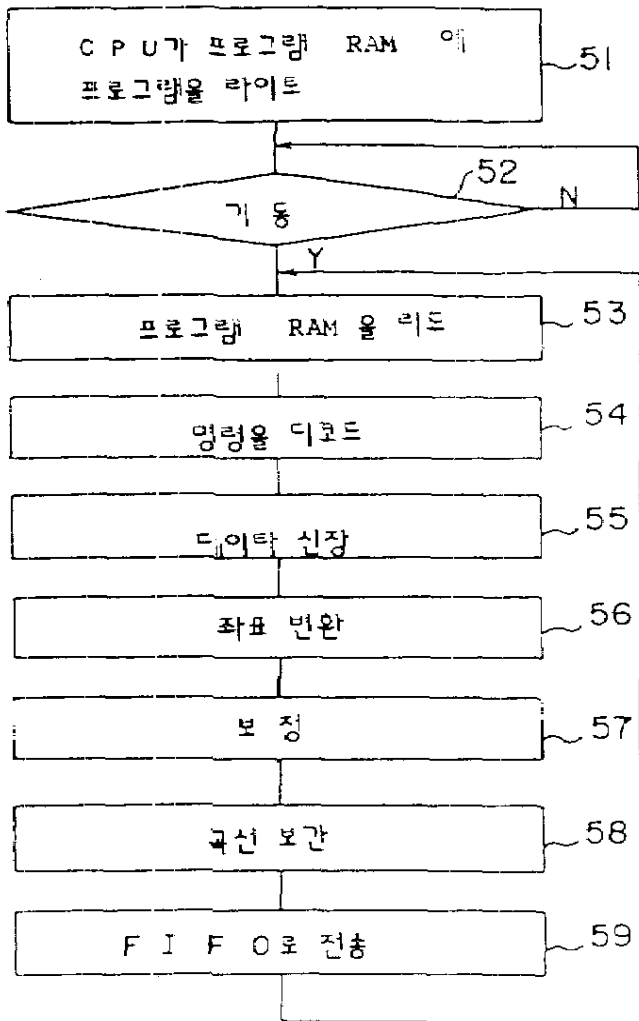
도면6



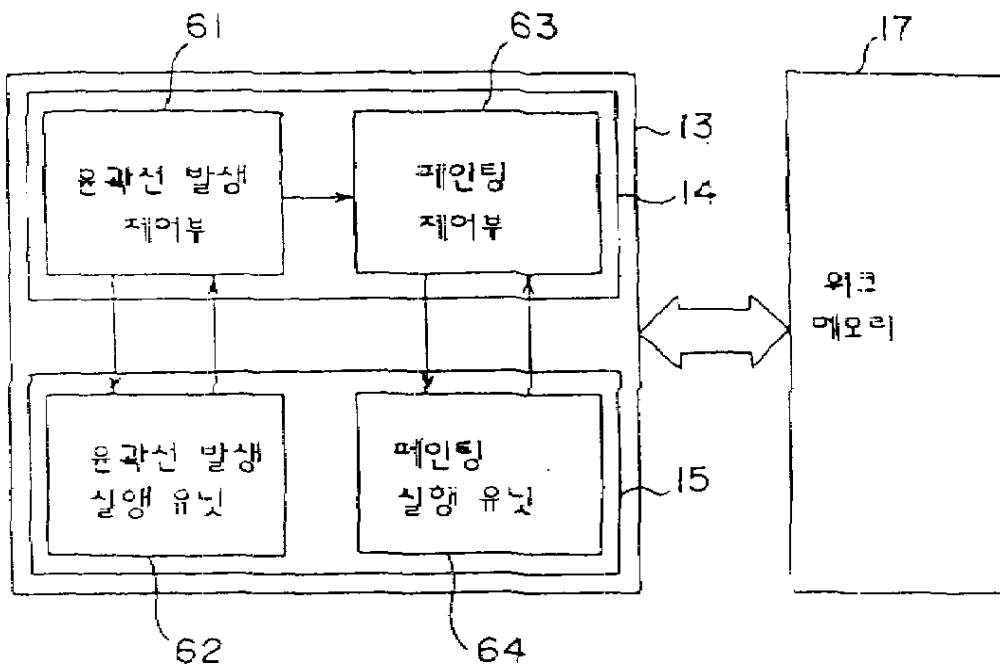
도면7



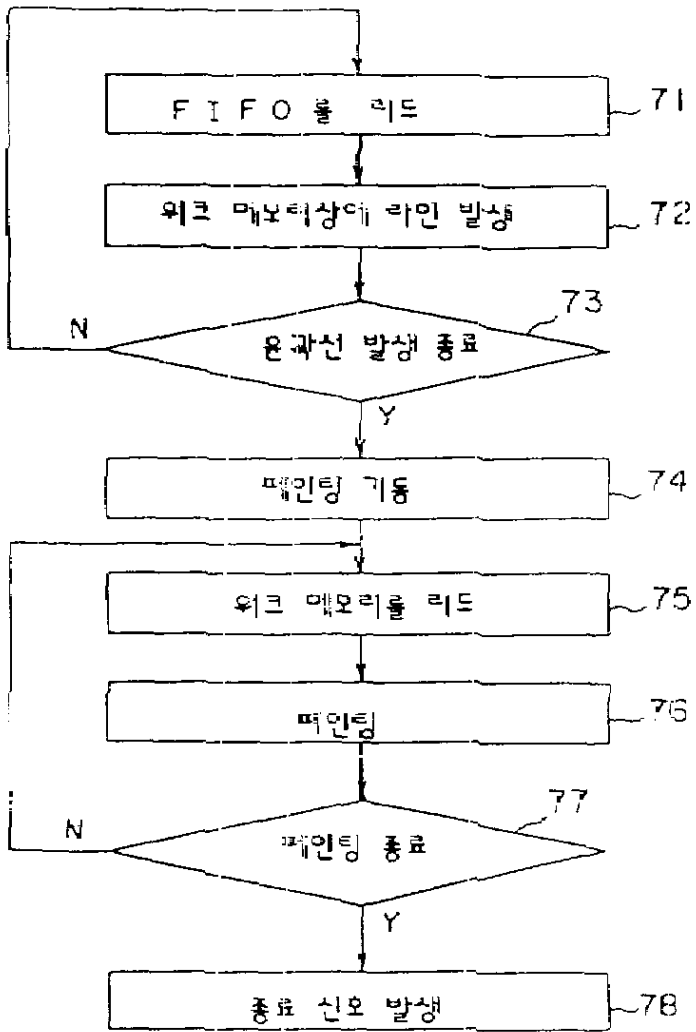
도면8



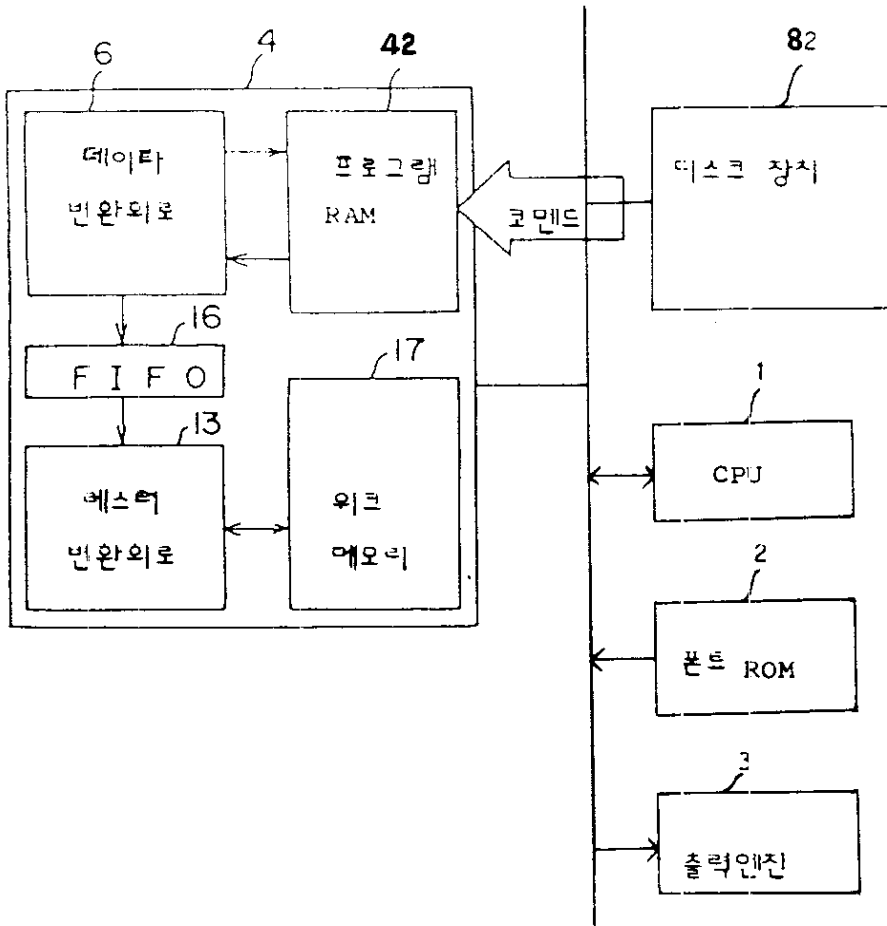
도면9



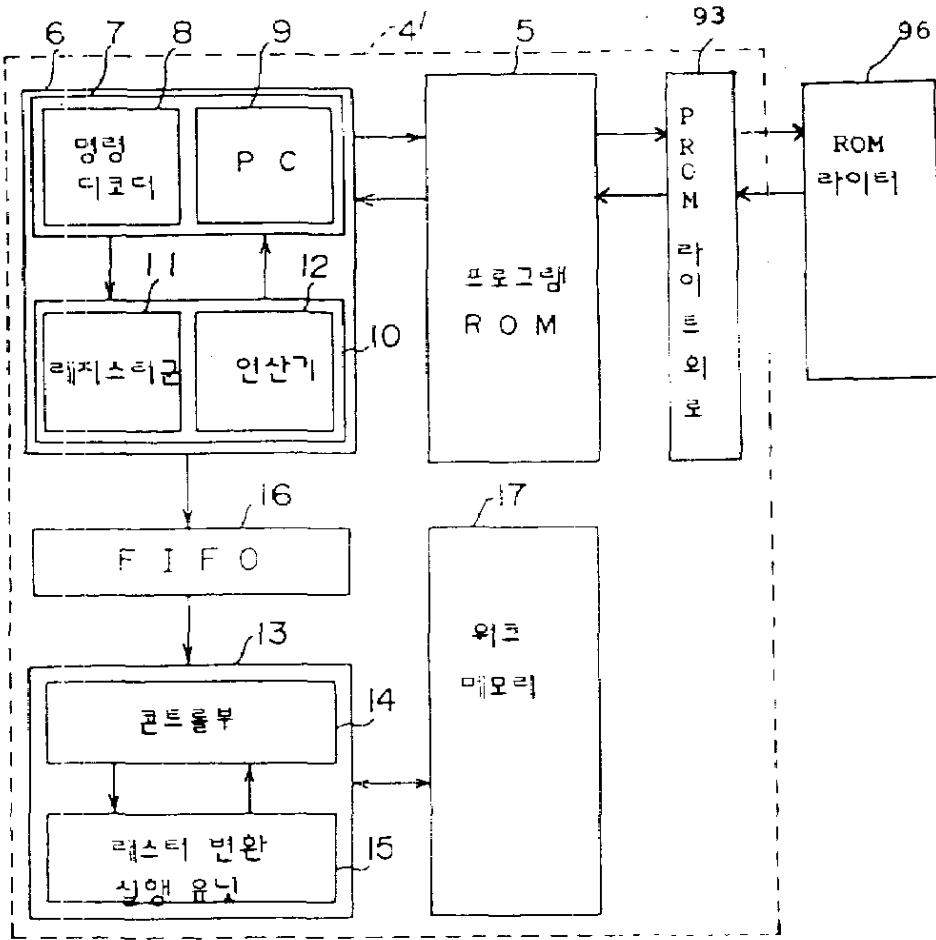
도면10



도면11

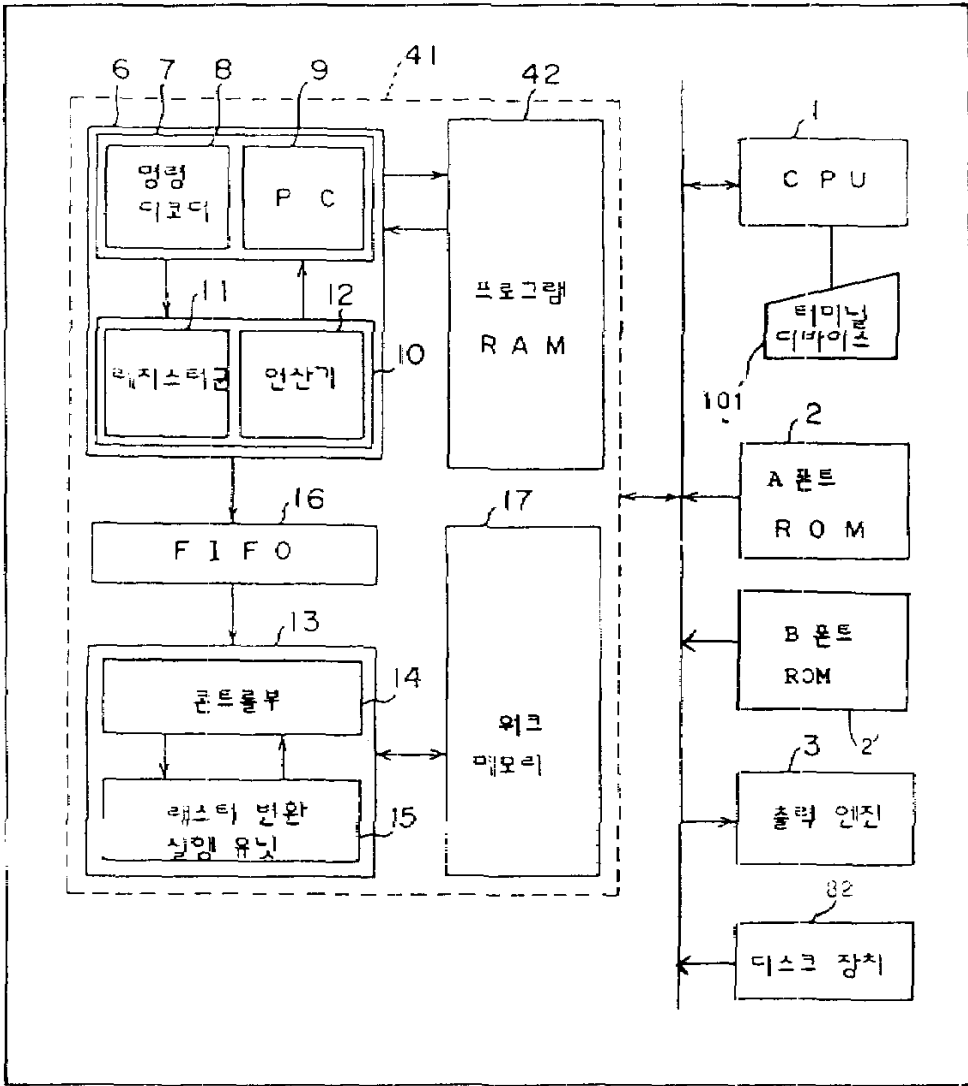


도면12

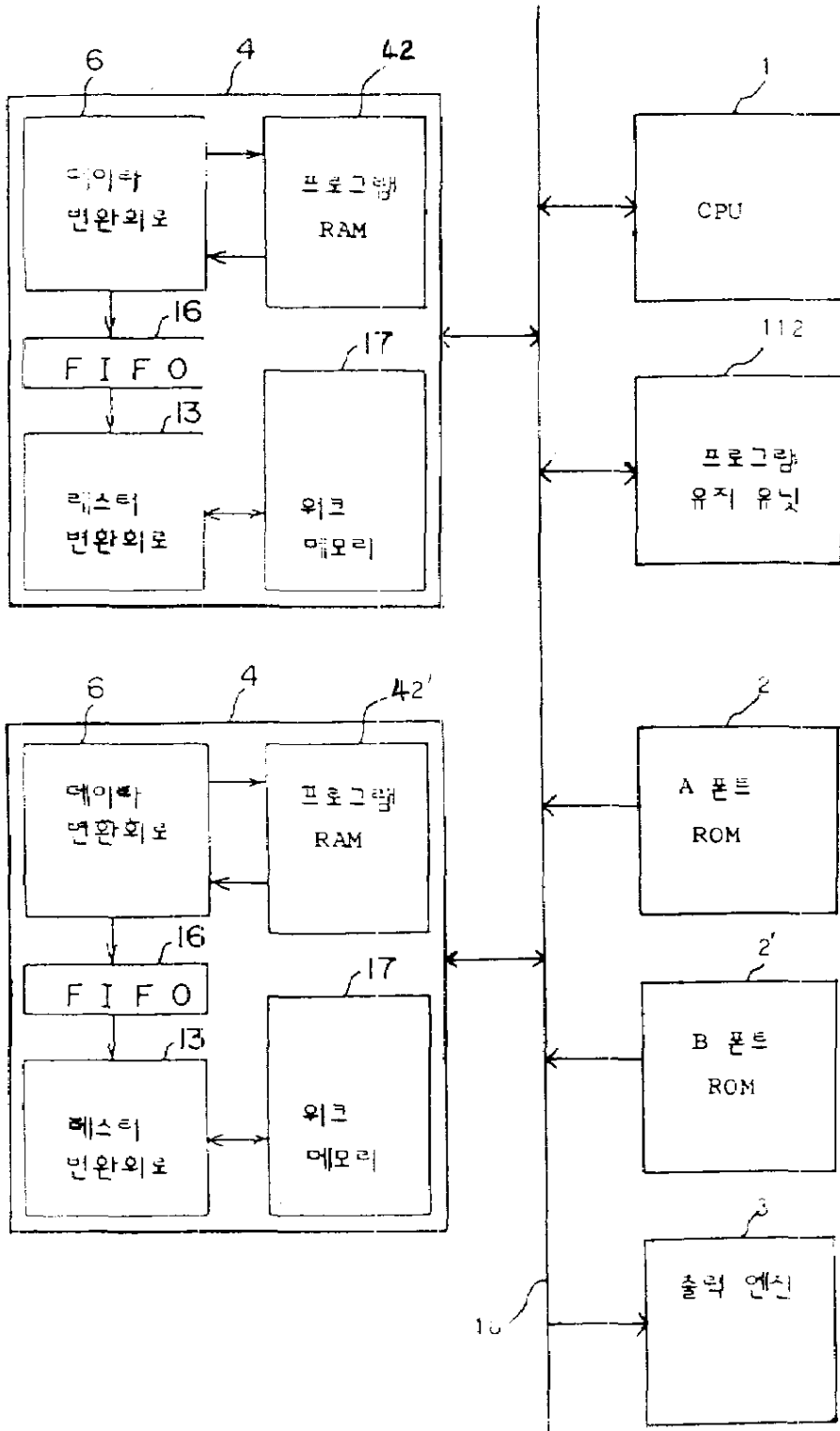


도면13

102



도면14



도면15

