(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) Int. Cl.[6]
G06F 3/03

(45) 공고일자 1998년12월15일
(11) 등록번호 특0160801
(24) 등록일자 1998년08월20일

(21) 출원번호 특1994-040321
(22) 출원일자 1994년12월31일
(65) 공개번호 특1995-027547
(43) 공개일자 1995년10월18일
(30) 우선권주장 8/210,610 1994년03월18일 미국(US)

(73) 특허권자 인터내셔널 비지네스 머신즈 코포레이션 윌리엄 티. 엘리스
미합중국 10504 뉴욕주 아몬크
(72) 발명자 랜달 엘. 버트램
미합중국 40515 켄터키주 렉싱톤 메도우뷰 드라이브 1524
제임스 엘. 콤스
미합중국 40515 켄터키주 렉싱톤 부룩크셔 서클 2417
(74) 대리인 장수길

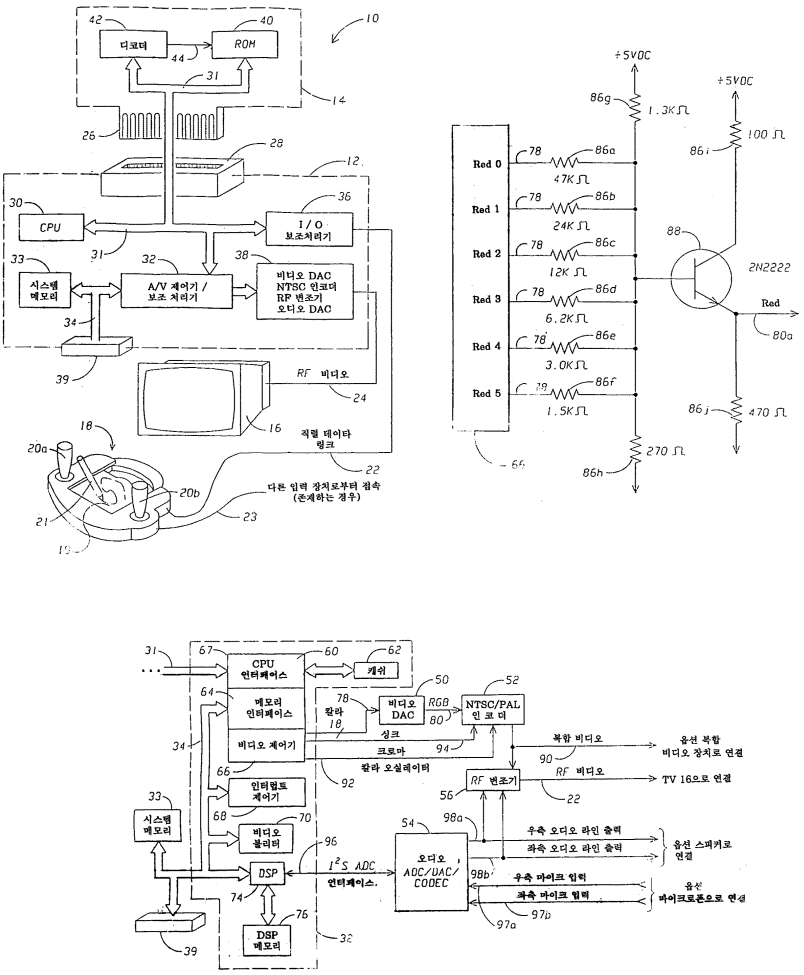
심사관 : 박제현

(54) 운영 체제에서 터치패드를 지원하는 컴퓨터 시스템

요약

터치패드(touchpad) 등으로부터 좌표 타입 데이터(coordinate type DATA)를 받아 들이도록 구성된 중앙 처리장치(CPU)를 포함하는 컴퓨터 시스템이 제공된다. CPU는 터치패드에 인터페이싱(interfacing)하기 위해 특별한 지원을 하는, CPU 상에서 실행되는 운영 체제를 가진다. 운영 체제는 다음 성능을 지닌다. (1) 터치패드의 기하학적 영역을 매핑하고 상기 영역을 응용 프로그램에 응답하는 특정 영역 식별자(specific region identifier)에 할당한다. (2) 접촉된 영역의 영역 식별자를 결정하고, 상기 영역 식별자를 응용 프로그램에 전달한다. 영역을 정의하는데 사용되어진 명령들의 유니트를 변화시키기 위한 지원도 제공된다.

대표도



명세서

[발명의 명칭]

운영 체제에서 터치패드를 지원하는 컴퓨터 시스템

[도면의 간단한 설명]

제1a도와 제1b도는 본 발명의 시스템의 일반 레이아웃을 보여주는 블록도.

제1c도는 본 발명의 시스템에서 사용된 비디오 디지털/아날로그 변환기의 상세한 회로도.

제2a도는 디폴트 템플리트(default template)를 가진 터치패드(touch pad), 두 개의 조이스틱(joystick) 및 템플리트 오버레이(template overlay)를 나타낸 본 발명의 입력 장치의 평면도.

제2b도는 템플리트 오버레이 리테이너(retainer)에 삽입된 템플리트 오버레이를 갖는 본 발명의 입력 장치의 평면도.

제2c도는 제2b도에 나타낸 본 발명의 입력 장치의 일부분의 확대 평면도.

제2d도는 제2c도의 2D-2D 라인으로 절단한 단면을 도시한 단면도.

제2e도는 제2c도의 2E-2E 라인으로 절단한 단면을 도시한 단면도.

제2f도는 제2a도의 2F-2F 라인으로 절단한 단면을 도시한 단면도.

제2g도는 제2a도에 도시한 본 발명의 입력 장치의 일부분을 절개한 확대 평면도.

제2h도~제2k도는 식별 패턴(identification patterns)을 나타낸 템플리트 오버레이 단부(edge)의 각 실시예의 저면도.

제2l도는 본 발명의 입력 장치의 정면도.

제2m도는 본 발명의 입력 장치에서 사용되는 조이스틱을 상세히 도시하고, 제2a도의 2m-2m라인으로 절단한 단면을 도시한 부분 단면도.

제3도는 본 발명의 입력 장치의 전기 회로를 나타낸 블록도.

제4도는 본 발명의 터치패드를 지원하는 운영 체제에서 매핑(mapped)된 터치패드의 구성의 한 실시예를 도시한 도면.

* 도면의 주요부분에 대한 부호의 설명

12 : 데이터 프로세싱 유니트 14 : 프로그램 카트리지

16 : 표준 텔레비젼 세트 19 : 터치패드

30 : 중앙 처리 장치 32 : A/V 제어기/코프로세서

33 : 시스템 메모리

50 : 비디오 디지털 / 아날로그 변환기(video DAC)

52 : NTSC/PAL 인코더 60 : 프로세서 인터페이스

74 : 디지털 신호 처리기(DSP) 76 : DSP 메모리

90 : 복합 비디오 신호 102 : 템플리트 오버레이

110 : 패드 표면 122 : 터치패드 센서

124 : 오버레이 센서 202 : 좌표 결정 회로

210 : 인터페이스 회로

[발명의 상세한 설명]

본 발명은 디지털 시스템에서 일반적으로 이용되는 좌표 타입의 포인팅 장치(coordinate-type pointing device)에 관한 것이고, 특히 운영 체제에서 터치패드를 지원하는 컴퓨터 시스템에 관한 것이다.

비디오 그래픽 컴퓨터 시스템은 잘 알려진 대중적인 소비재 제품이다. 전형적인 비디오 그래픽 컴퓨터 시스템은 게임의 또는 다른 응용 프로그램의 영상을 디스플레이하기 위해 일반 TV 수상기에 연결되는 데이터 프로세싱 유니트를 포함한다. 데이터 프로세싱 유니트는 통상 카트리지 형태로 패키지되어 있는 롬(ROM)으로부터, 제어 소프트웨어(controlling software)를 제공받는다. 카트리지는 착탈 가능하게(removably) 데이터 프로세싱 유니트에 삽입되어(plugged into) 사용된다. 사용자가 제어 소프트웨어에 의한 응용 실행에 사용되는 위치 정보를 입력할 수 있도록 하기 위해 마우스, 조이스틱, 터치패드, 터치스크린, 스위치 패드 또는 광선총(light gun)등의 하나 이상의 포인팅 장치가 데이터 프로세싱 유니트에 연결된다.

데이터 프로세싱 유니트는 통상적으로 하나의 CPU, 관련 휘발성 및 비휘발성 메모리 (모든 램(RAM), 부트스트랩 ROM (부트 ROM)을 포함), 텔레비젼 (RF 비디오) 신호 발생기 및 여러 가지 포인팅 장치에 인터페이스하기 위한 입출력 프로세서(I/O processor)로 구성되어 있다. 이 장치들은 회선을 통해 통신을 한다(incircuit communication). 이 시스템들의 독특한 특징 중의 하나는 본체 기판(motherboard) 또는 상기 부품들의 전기적 상호 연결을 위한 평면적 시스템을 사용하는 점에 있다.

터치패드는 컴퓨터 시스템 좌표 타입의 데이터를 입력하는데 쓰이는 좌표 타입의 포인팅 장치이다. 터치패드는 전형적으로 그 표면에서 국부적 압력(localized pressure)를 검출할 수 있는 압력에 감응하는 경계 평면(Pressure-sensitive bounded plane)이다. 사용자가 터치패드의 표면을 손가락, 철필 또는 이와 유사한 것으로 접촉하면, 터치패드는 접촉된 위치를 결정하고, 그 위치의 좌표를 어떤 형태의 발생된 신호를 통해 부착된 컴퓨터 시스템에 보고한다. 이에 대하여 컴퓨터는 접촉된 위치와 관련된 기능이 있는 경우 그 기능을 수행함으로써 응답하게 된다.

보통 터치패드의 하나 혹은 2 이상의 영역은 데이터를 입력하거나 방향 입력을 제공하는 등 시스템내 또는 응용 프로그램 내에서 어떤 기능을 할당받게 된다. 사용자는 종래에는 템플리트에 의해 어떤 영역이 어떤 기능과 관련되는지 알 수 있었다. 템플리트는 그래픽 디자인된 시트(sheet)로서 보통 터치패드 표면 위에 포개져 있다. (superimposed). 그래픽 디자인은 보통 터치패드의 표면 영역을 표시하거나 매핑하고, 어떤 기능이 다양하게 매핑된 영역중 어느 영역과 관련되는지 사용자에게 상기 시켜 주기 위해 그 영역을 라벨로 표시하게 된다.

전형적인 터치패드 시스템에서, 응용 프로그램은 터치패드로부터 좌표 위치 데이터 신호를 받는다. 예를 들면, 터치패드가 패드 표면이 15번째 행과 12번째 열이 만나는 지점에서 접촉된 것을 통신하면, 응용 프로그램은 그 좌표 위치와 그 특정 영역에 매핑된 기능을 서로 관련시켜야 한다. 즉, 응용 프로그램은 그래픽 디자인 영역이 접촉된 것을 나타내는 신호를 해독하고(decode), 이에 따라 접촉된 위치와 관련된 기능을 결정해야만 한다.

각 응용 프로그램에서 어떤 영역이 접촉되었는지를 결정하는 것에는, 몇가지 문제점이 있다. 먼저, 응용 프로그램은 하드웨어에 종속하게 된다. 특정 응용 프로그램은 해상도(열과 행의 수), 전체 사이즈, 그리고 사용된 터치패드의 데이터 포맷을 알아야만 하고, 따라서 그렇지 않으면 응용 프로그램이 터치패드의 다양한 형태에 따라 실행되지 않는 경우도 발생하기 때문이다. 기술이 진보하여 터치패드의 해상도가 증가함에 따라, 현존 프로그램은 그것에 적응할 만한 유연성(flexibility)이 없을 수도 있다.

둘째로, 이에 따라 응용 프로그래머들이 응용 프로그램에 위치 해독 코드를 기입해야 한다는 점이다. 그래서, 각 응용 프로그램은 고유의 영역 결정 루틴을 가져야만 하고 그로 인하여 불필요한 노력이 중복되고 터치패드 영역이 지원되는 방식에 있어서도 불일치와 에러까지도 유발할 수 있다.

본 발명에 있어서, 컴퓨터 시스템은 운영 체제에서 터치패드의 지원을 제공한다. 운영 체제는 다음과 같

은 능력을 가진다.

(1) 터치패드의 기하학적 영역을 매핑하고 응용 프로그램에 응답하는 특정한 영역 식별자(region identifier)에 각 영역을 할당하는 능력.

(2) 접촉된 영역의 영역 식별자를 결정하고, 응용 프로그램에 영역 식별자를 전달하는 능력.

이러한 통상의 능력은 터치패드에 인터페이싱하는 작업에 있어서 일관성과 유연성을 부여한다.

본 발명의 여타의 이점은 이하의 상세한 설명에 의하여 더 명백하게 설명될 것이다.

제1a도 및 제1b도에는 본 발명의 컴퓨터 시스템(10)이 도시되어 있다. 제1a도에서 도시된 바와 같이 시스템(10)은 그것에 착탈 가능하게 접속된 프로그램 카트리지(14)가 구비된 데이터 프로세싱 유니트(12)를 포함한다. 또한, 데이터 프로세싱 유니트(12)에는 표준 TV세트(16)와 입력 장치(18)이 연결되어 있다. 터치패드(19)와 2개의 조이스틱(20a, 20b)를 갖는 입력 장치(18)은 사용자의 손가락, 철필(21) 또는 이와 유사한 것이 터치패드(19)상에 접촉하는 위치에 해당하는 좌표 타입의 데이터를 데이터 프로세싱 유니트(12)에 전송한다. 또한, 입력 장치(18)은 조이스틱(20a, 20b)의 움직임에 대응하는 방향 지정 형태의 데이터를 데이터 프로세싱 유니트(12)에 전송한다. 제1a도에 나타내지는 않았지만, 표준 TV세트(16)은 복합 비디오 신호를 받아들일 수 있는 한 쌍의 스피커와 디스플레이 장치로 대체될 수 있다. 입력 장치(18)은 직렬 데이터 링크(22)를 거쳐 데이터 프로세싱 유니트(12)와 연결되어 있다. TV세트(16)은 데이터 프로세싱 유니트(12)에 RF 비디오 라인(24)를 통하여 접속되어 있다.

카트리지(14)는 엣지 카드 접속기(26)을 갖는데, 이 엣지 카드 접속기(26)이 카트리지 접속기(28)에 연결되어 카트리지(14) 내의 장치를 데이터 프로세싱 유니트(12) 내의 장치에 전기적으로 연결한다.

데이터 프로세싱 유니트(12)는 SYSTEN 버스(31)과 연결된 CPU(30)과 오디오/비디오(A/V) 제어기/코프로세서(32), SYSTEM 버스(31)로부터 A/V 제어기/코프로세서(32)에 의하여 생성되는 SYSTEM 버스(34)에 연결되어 있는 시스템 메모리(33), 제1 및 제2 디코더 칩(도시되지 않음), 입출력 코프로세서(36), 두 개의 카트리지 접속기(하나는 28, 다른 하나는 도시되지 않음), 오디오 및 비디오 신호를 생성시키는데 필요한 부가적 회로(38) 및 확장 접속기(39)로 구성되어 있다. 이들 장치는 도면에 도시된 바와 같이 회선으로 통신을 하도록 연결되어 있다. 부가적 회로(38)은 제1b도에 도시되어 있고, 제1b도에 관한 설명에서 더욱 상세히 기술하고 있다.

CPU(30)은 당해 분야에서 잘 알려진 바와 같이 DATA 버스, ADDRESS 버스와 CONTROL 버스의 다중 버스를 발생시킨다. 위의 세 버스들을 집합적으로 SYSTEM 버스(31)이라 칭한다. 바람직한 실시예에 있어서 CPU(30)은 95051 미합중국 캘리포니아주 산타클라라시 바우어스 애비뉴 3065 소재 (Intel)사의 80376이다. 80376은 인텔사의 잘 알려진 80386 SX의 하나의 변형이며, 80376은 16비트 모드가 아니고, 32비트 모드에서 시동된다는 점에서 80386 SX와는 다르다. 특히, 0비트를 로직 ONE으로 만들어서 CR0 레지스터는 0011h(0011은 16진수)상태로 세트되므로, 376이 32비트 메모리 모드에서 효과적으로 동작하게 된다. 페이징(paging)은 가상 386 동작을 허용하도록 인에이블된다.

A/V제어기/코프로세서(32)는 sysstem 버스(31)로부터 세 개의 예비 범용 입출력 디코더 라인(GPI01, gp102, gp103)을 생성하며, 각각의 디코더 라인은 32비트의 입출력 어드레스 범위(i/o address range)를 제공한다. 범용 디코더는 A/V 제어기/코프로세서(32) 외부의 장치에 3개의 액티브 로우 칩 인에이블(active low chip enable)을 제공하는 데 사용될 수 있다. 데이터 프로세싱 유니트(12)에 있어서, 범용 디코더는 입출력 코프로세서(36; GPI101)과 2개의 카트리지 커넥터(GPI02, GPI03)에의 어드레tm 영역을 디코드하는데 사용된다. A/V 제어기/코프로세서(32)의 나머지 회로는 이하에 설명한다.

시스템 메모리(33)은 스크린 RAM, 시스템 RAM 및 부트스트렘 ROM(모두 도시안됨)을 포함한다. 온 보드 스크린 RAM(on-board screen RAM)과 시스템 RAM은 1MB 크기의 32비트 DRAM이다. 적합한 DRAM은 32비트의 메모리를 실현하도록 구성된 도시바 사의 16비트 메모리 칩에 의한 한 쌍의 TCS14170BJ 256 킬로바이트이다. CPU(30)의 어드레스 공간의 한 부분은 A/V 제어기/코프로세서(32)내의 다수의 8비트 레지스터에 디코드된다. 모든 내부 위치는 짝수 어드레스 경계에 놓이게 되며, 워드-폭 입출력(word-wide I/0) 읽기와 쓰기는 적합한 곳에 수행될 수 있다. 이 특별한 실시예에서, 바이트-폭(byte-wide) 쓰기는 워드-폭 레지스터에서 수행될 수 없고, 입출력 사이클은 홀수 어드레스를 액세스하는데 사용될 수 없다. 부트스트랩 ROM은 항상 16비트 폭이다. 부트스트랩 ROM은 여러 제조 회사에 의해 제조된 2개의 27C512 지울 수 있는 프로그래머블 ROM(EPROM)으로 구성되어 128K의 부트스트램 ROM이 주어진다. 리셋을 하면 16MB의 어드레스 범위를 통하여 ROM 및 내부 메모리를 포함하는 F20000H로부터 FFFFFFH까지의 1MB 윈도우가 반복된다.

시스템 메모리(33)은 다수의 장치 사이에 공유된다. A/V 제어기/코프로세서는 시스템 메모리(33)을 위한 중재자(arbitrator)이고, 따라서 SYSTEM 버스(31)은 A/V 제어기 / 코프로세서(32)에 의해 SYSTEM버스(34)로 변경된다.SYSTEM버스(34)는 도시되어 있지 않은 DATA'버스, CONTROL'버스로 구성되어 있다. 따라서, 시스템 메모리(33)은 SYSTEM'버스(34)를 통하여 억세스 된다.

입출력 코프로세서(36)은 CPU(30)을 입력 장치(18) 및 도시되지 않은 키보드, 콘트롤러, 마우스, 프린터와 같은 옵션 장치등의 여러 입력 장치에 인터페이스 시키게 된다. 바람직한 실시예로서, 입출력 코프로세서(36)은 2MHz로 동작하는 모터롤러사에 의해 제조된 미리 프로그램된(preprogRAMmed) MC68HC705C8(이하, 68HC7 05라 한다)를 사용할 수 있다. 68HC705 입출력 코르로세서(36)은 68HC705를 주변 장치로 하여 구성함으로써 CPU(30)에 인터페이스된다. (1) PA0-PA7은 DATA버스의 D0-D7에 접속시킨다. (2) PB7, PB1과 PB2는 GPI01(A/V 제어기/코프로세서(32)에 의해 디코드된 32-바이트 어드레스 영역, 후술하는 바와 같음), CONTROL 버스 및 ADDRESS 버스의 A1, A2에 각각 접속시킨다. (3) PB3, PB4와 PB5는 CONTROL 버스의 ADS, READY, W/R에 각각 접속시킨다. 입출력 코프로세서(36)은 A/V 제어기/코프로세서에 의하여 디코드되어 입출력 스페이스의 네 개의 16비트 어드레스(이하, AS0, AS2, AS4, AS6라 한다)를 갖는다.

68HC705 안에 있는 프로그램은 CPU(30)에 다음 방식으로 인터페이스된다. 68HC705는 프로세서 버스(processor bus)에 직접 접속되도록 설계되며, CPU(30)의 입출력 포트(I/O port)로 작용한다. 한 쌍의 내

부 래치는 다른 프로세서가 수신할 준비가 될 때까지 각 프로세서 사이를 통과하는 데이터를 보존한다. 각 프로세서의 상태 비트(status bit)는 데이터 래치의 상태를 보여준다. 상태 비트를 체크함으로써 이전 데이터(previous DATA)가 읽혀졌는지 혹은 새로운 데이터가 읽혀지기를 기다리고 있는 여부를 알 수 있다.

I/O 코프로세서(36)은 특히 다음과 같은 기능을 실행한다 : (1) 50ms 타이머, (2) 입력 장치로부터 통신 패킷을 수신하는 직력 제어기 링크, (3) 각 카트리지 접속기에 카트리지(14)의 존재 유무와 확장 접속기에 확장 장치 또는 CD 드라이브의 존재 유무를 판단하는 카트리지/확장 감지, (4) 시스템 리셋, 및 (5) I^2C 비휘발성 RAM(NVRAM) 인터페이스. 상기I/O 코프로세서(36)은 또한 선택적인 CD 드라이브와 함께 통신할 수 있도록 선택적인 DSA 콤팩트 디스크 제어 직력 라인을 실행한다.

상기 50ms 타이머는 규칙적으로 50밀리초 간격으로 종료하는 68HC705 I/O 코프로세서(36)의 와치도그 타이머(watchdog timer)를 구성함으로써 구현된다. 와치도그 타이머가 종료할 때마다, 상기 I/O 코프로세서(36)는 A/V 제어기/코프로세서(32)(A/V 제어기/코프로세서는 AIO 로우(low)를 풀(pulling)하는 I/O 코프로세서에 응답하여 IRQ 라인을 거쳐 CPU를 인터럽트한다)의 아날로그 인터럽트 0(AI0)를 사용하여 CPU(30)를 인터럽트한다. 상기 CPU는 I/O 포트 AS0에 바이트 0F0H 또는 바이트 00H를 각각 기록함으로써 50ms 타이머를 인에이블하고 디스에이블한다. 상기 타이머는 디폴트인 경우 인에이블한다.

CPU의 인터럽트 확인 응답 사이클 동안, 상기 A/V 제어기/코프로세서는 인터럽트 처리 루틴의 어드레스를 어써트(assert)한다. 상기 인터럽트 처리 루틴은 CPU(30)으로 하여금 I/O 코프로세서에 대응하는 16비트 I/O 포트 AS0로부터 한 개 이상의 바이트를 읽게 한다. 각각의 I/O 포트 AS0동안, 상기 A/V 제어기/코프로세서(32)로 하여금 I/O 코프로세서(36)을 선택하게 함으로써, CPU(30)과 I/O 코프로세서(36) 간에 데이터 전달을 할 수 있게 한다.

상기 I/O 코프로세서(36)는 50ms 인터럽트에 응답하여 CPU로 전달된 한 바이트를 항상 가지게 될 것이다. 이 바이트의 하위 니블(lower nibble)은 마지막 인터럽트 확인응답 사이클과 이 바이트의 상위 니블이 CPU로 전달될 I/O 장치 메시지의 수를 포함하기 때문에 50ms 시간의 종료 번호를 포함한다. 만약 50ms 타이머가 디스에이블되면, 이때 이 바이트의 하위 니블은 제로일 것이다. 만약 15이상의 메시지가 수신되면, 이 때 15는 상위 니블로 보내지고, 임의의 나머지 메시지는 다음 전송 동안 보내진다. 이와 같은 제 1 바이트의 내용에 따라, CPU는 I/O 코프로세서(36)로부터 다음 바이트를 판독하는데, 이것은 대부분 입력 장치에서 오는 데이터 패킷일 것이다. 전형적으로, 상기 입력 장치느 그 각각의 상태가 변할 때만 전달됨으로써, 메시지 전송 주파수를 극히 낮게 유지한다.

상기 입력 장치(18)과 다른 모든 입력 장치는 직렬 데이터 링크(22)를 거쳐 I/O 코프로세서(36)에 접속되어 있다. 각각의 입력 장치(예를들어, 입력장치 18)은 직렬 링크(22)를 따라 전송하는데 적합한 포맷으로 제어 장치의 이동을 변형시킨다. 상기 입력 장치(18)는 직렬 데이터 링크(22)를 거쳐 데이터 프로세싱 유니트(12)로 데이터 패킷을 전달한다. 이하에서 설명된 바와 같이, 데이터 패킷의 구조는 입력 장치의 종류에 따라 다르다. 좌표 타입 장치(마우스, 아날로그 조이스틱, 터치패드 등)은 스위치폐쇄 형태 장치(키보드, 디지털 조이스틱, 스위치 패드 등) 과는 다른 데이터 패킷 구조를 갖는다.

직렬 제어기 링크(22)는 세 개(3)의 라인 : 데이터 수신 라인, VCC (+5 VDC)라인, 및 접지 라인으로 구성 되어 있다. 68HC705는 68HC705의 PD0/RDI핀을 사용하여 제어기 직렬 링크의 데이터 수신 라인을 구현한다. 이 핀은 공지된 비동기 포맷을 사용하여 직렬 장치에의 인터페이스로써 사용되도록 설계되어 있다. 직렬 전송은 다음과 같은 포맷을 갖는다 : 초당 4800피트, 노 패리티(no parity), 8 데이터 비트, 및 한 개의 정지 비트. 클릭된 동기 포맷은 선택적으로 사용될 수 있다. 직렬 제어기 링크(22)는 기술분야에 공지된 6-컨덕터 미니-딘(mini-din) 플러그 접속기 (도시안됨)에 접속되어 있다. 입력 장치는 연쇄적으로(daisy-chained) 연결되어 있으므로, 단일 장치가 데이터 프로세싱 유니트(12)에 접속되게 된다. 예를 들면, 만약 소위 마우스 포인팅 장치가 시스템(10)에 부가되어 있으면, 상기 마우스는 프로세싱 유니트(12)에 접속되어 있는 장치(18)에 접속되어 있다.

카트리지 감지 및 확장 감지는 각 카트리지 접속기 또는 확장 접속기에서 카트리지(14)의 존재 유무를 판단하기 위한 것으로, I/O 코프로세서(36)로 하여금 카트리지 접속기(28)의 핀을 풀하게 함으로써 구현된다. 상기 핀은 시스템 플래너(planar)상의 적당한 풀업 저항(도시안됨)에 의해 논리 ONE으로 풀되고, 적절히 접속된 카트리지(14)는 상기 핀을 논리 ZERO로 풀한다. 그래서, 각 카트리지 감지에서의 ONE은 카트리지(14)가 존재하지 않음을 가리키고, ZERO는 카트리지(14)가 존재한다는 것을 가리킨다. 이와 유사하게, 확장 감지에서의 ONE은 선택적인 CD드라이브와 같은 확장 장치가 존재하지 않음을 가리키고, ZERO는 확장 장치가 존재함을 가리킨다.

리셋은 A/V 제어기/코프로세서(32)의 리셋 신호에 대한 I/O 코프로세서(36)을 제어함으로써 실행되는데 상기 I/O 코프로세서는 CPU(30)의 리셋 신호를 제어한다. CPU(30)은 I/O 코프로세서(36)로 하여금 CPU(30)를 리셋시키는 A/V 제어기/코프로세서를 리셋하게 함으로써 시스템(10)을 리셋하도록 I/O 코프로세서(36)에 명령한다. CPU(30)는 I/O 코프로세서(36)로 하여금 시스템 리셋을 발생시켜 I/O 포트AS0에 바이트 0FFH를 기록하게 한다. 또한, 상기 I/O 코프로세서(36)은 시스템에 대한 선택적인 리셋 스위치(도시되어 있지 않음)를 모니터하고, 스위치 폐쇄를 검출 할 때 시스템을 리셋한다.

결과적으로, 상기 I/O 코프로세서는 비휘발성 시스템 RAM의 512 바이트 내용을 판독, 기록, 및 확인하기 위해 I^2C 비휘발성 RAM(NVRAM) 인터페이스를 실행한다. 상기 NVRAM(도시되어 있지 않음)은 필립스 반도체(주)에 의해 제조된 PCF8594를 구비하고 I^2C 인터페이스를 거쳐 I/O 코프로세서와 회선으로 통신한다. 보다 큰 NVRAM 용량을 제공하기 위해 한 개 이상의 PCF8594가 다단으로 접속될 수 있다. NVRAM을 액세스하기 위해, 3-바이트 시퀀스가 사용된다. 3 바이트 모두는 I/O 포트 AS0를 거쳐 액세스된다. CPU에 의해 I/O 코프로세서에 기록되는 제 1 바이트는 전송이 판독인지 또는 기록인지를 가리키고, I/O 코프로세서에 세그먼트 어드레스를 부여한다. 이 바이트의 하위 니블은 전송 형태를 가리키는데, 01H는 NVRAM으로부터의 기록을 가리키고, 02H는 NVRAM으로부터의 판독을 가리킨다. 이 바이트의 상위 니블은 NVRAM의 256-바이트 세그먼트에 대응하는 4-비트 세그먼트 번호이다. NVRAM의 512 바이트로써, 단지 밑부분의 두 세그먼

트(0 및 1)만이 사용된다. 판독 및 기록을 위해, 다음 바이트는 동일하고, 다음 바이트는 CPU에 의해 기록되고, 세그먼트내에서 액세스되는 바이트의 어드레스이다. 마지막 바이트는 CPU에 의해 I/O 코프로세서에 기록되거나 또는 상기 코프로세서로부터 판독되고, NVRAM에 기록되거나 또는 NVRAM으로부터 판독되는 데이터 바이트이다.

선택적으로, 상기 I/O 코프로세서는 다른 방법으로 구현될 수 있다. 예를들면, 3단의(tristated) 판독가능한 시프트 레지스터는 직렬 데이터 링크(22)로부터 정보를 적절히 수신 할 수 있다. 이 경우, CPU(30)는 상기 시프트 레지스터를 주기적으로 판독하여 입력 장치로부터 데이터 패킷을 액세스한다.

제1 디코드 칩(도시되어 있지 않음)은 CPU(30), A/V 제어기/코프로세서(32), 및 두 개의 카트리지 접속기(28)(그 외는 도시되어 있지 않음)과 전기 회선으로 통신한다. 상기 제 1 디코드 칩은 입력으로 SYSTEM 버스(31)의 상위 두 어드레스 라인을 수신하고, 80376 CPU(30)의 16 메가바이트 어드레스 공간을 두 개는 카트리지 접속기(28)(다른 것은 도시되어 있지 않음)을 위한 것이고, 하나는 A/V 제어기/코프로세서(32)를 위한 것인 3개의 칩 선택 라인으로 표현된 네개의 4 메가바이트 영역으로 디코드한다. 상기 상위 4 메가바이트와 하위 4 메가바이트는 A/V제어기/코프로세서 칩 선택으로 디코드되고, 나머지 두 개의 4-메가바이트 영역은 두 개의 카트리지 접속기 칩 선택으로 디코드된다.

제 2 디코더 칩(도시되어 있지 않음)은 확장 접속기(39)에 대한 칩 선택을 실행하는데 사용된다. 상기 제 2 디코드 칩은 A/V 제어기/코프로세서(32)와 SYSTEM' 버스(34)를 다른 확장 접속기(39)와 회선으로 통신한다. 상기 제2 디코드 칩은 A/V 제어기/코프로세서(32)로 하여금 F20000H에서 시작하는 시스템 ROM의 128K 블록을 디코드하게 한다. F40000H FFFFFFH까지의 범위는 확장 접속기(39)에 의해 사용되기 위해 상기 제2 디코드 집에 의해 디코드된다. 제 2 디코드 칩에 의해 디코드되는 ROM의 이 블록은 확장 접속기(39)를 거쳐 시스템(10)에 ROM을 부가하는데 사용된다.

또한, 데이터 프로세싱 유니트(12)는 카트리지(14)가 CPU(30)과 다른 시스템 구성요소와 회선으로 통신할 수 있도록 한쌍의 카트리지 접속기(한개는 도면번호 28, 다른 하나는 도시되어 있지 않음)을 갖는다. 상기 카트리지(14)는 금으로 도금된 62-핀(31 콘덕터의 두 행) 엣지 카드 접속기(26)를 거쳐 데이터 프로세싱 유니트(12)의 접속기(28)에 접속되어 있다. 상기 프로세싱 유니트(12)는 엣지 카드 접속기(26)의 엣지 카드 접속을 수신하기 위한 두 개의 카트리지 접속기(28)을 갖는다. 상기 카트리지(14)는 접속기(28)의 컨덕터와 부합하기 위해 금으로 도금된 카드 엣지 접속을 가짐으로써, 카트리지(14)가 처리기 유니트(12)에 플러그식으로 접속될 수 있게 한다. SYSTEM 버스(31) 신호, 카트리지 감지 라인, 전원, 접지, 아날로그 인터럽트1 또는 2(각 카트리지는 독특한 인터럽트를 갖는다), GPIO 2 또는 3 (각 카트리지는 독특한 칩 선택을 갖는다), 록 라인(lock line : 이것은 80376 및 80386SX SYSTEM 버스 31의 전형적인 신호이다), 및 제 1 디코드 칩에 의해 발생되는 카트리지 선택과 같은 신호는 카트리지 접속기(28 : 다른 것은 도시되어 있지 않음)를 거쳐 외부 장치로 통신된다. 선택적으로, 부가적인 CD 드라이브에 접속하는데 필요한 신호는 또한 카트리지 접속기(28)를 거쳐 외부 장치에 접속될 수 있다.

또한, 프로세서 유니트(12)는 단일 112-핀(각각 56핀의 두 행) 엣지 카드 확장 접속기(39)를 갖는다. 상기 확장 접속기(39)는 장치로 하여금 시스템 메모리(33)에 보다 많은 메모리를 부가하게 하고 다른 여러가지 특징을 부가하게 된다. 상기 확장 접속기(39)에 접속된 장치는 확장 접속기와 일치하기 위해 금으로 도금된 카드 엣지를 가짐으로써, 장치로 하여금 프로세서 유니트(12)에 플러그식으로 접속될 수 있게 한다. SYSTEM'버스 신호, 확장 접속기(39) 감지 라인, 전원, 접지, CAS 및 RAS 라인, 및 제2 디코드 칩에 의해 발생되는 확장 접속기(39) 선택과 같은 신호는 확장 접속기(39)를 거쳐 외부장치로 통신된다. 선택적으로, 부가적인 CD 드라이브에 접속된 신호는 또한 확장 접속기(39)를 거쳐 외부 장치에 접속될 수 있다.

프로그램 카트리지(14)는 프로그램 ROM(40)과 디코더(42)를 구비한다. 선택적으로, 상기 디코더(42)는 프로세싱 유니트(12)내로 설계될 수 있다. 상기 프로그램 ROM(40)은 판독 전용 메모리 포맷으로 CPU(30)상에서 실행하는데 적합한 코드를 포함할 수 있다. 선택적으로, 베터리에 의해 지원되는 RAM(battery-backed RAM)과 같은 다른 메모리 유형이 카트리지(14)의 기억 장치로써 사용될 수 있다. 상기 프로그램 ROM(40)은 제1a도에 도시된 바와 같이 CPU(30)과 회선으로 통신한다.

카트리지(14)내의 어드레스 디코더(42)는 프로그램 ROM(40)에 적합한 메모리 범위에 대한 ADDRESS 버스의 전체 폭을 디코드하고, 기술분야에 공지된 바와 같이 ROM(40)에 의해 요구되는 칩 선택 신호(44)를 발생한다. 상기 어드레스 디코더(42)는 기술분야에 공지되어 있고, 많은 제조자 예를 들면, AMD(주)에 의해 제조된 16V8 프로그래머블 어레이 논리(PAL)에서 규현된다. 만약 디코더(42)가 프로세싱 유니트(12)에 설계되어 있다면, 이 때 선택(44)은 접속기(26)에 의해 ROM(40)에 전기적으로 통신된다.

지금부터 제1b도를 참조하면, 제1a도의 부수적인 회로(38)이 A/V 제어기/코프로세서(32)에 접속되어 있는 것으로 도시되어 있다. 상기 부수 회로(38)은 비디오 디지털 아날로그 변환기(비디오 DAC), NTSC/PAL (공지된 유럽 텔레비젼 신호 표준이라고 하는 PAL) 인코더(52), 오디오 디지털 아날로그 변환기/아날로그 디지털 변환기(ADC/DAC/CODE:54) 및 RF 변조기(56)인 4가지 장치를 구비한다. 각각은 도면에 도시된 바와 같이 접속된다.

오디오/비디오 제어기/코프로세서(A/V 제어기/코프로세서:32) 전자 회로는 ASIC(특정 응용 집적 회로)라고 하는 대용량의 주문형 논리 칩내에 포함되어 있다. 이하의 설명을 충족시키는 A/V 제어기/코프로세서(32)는 영국 MK9 1DP, 센트랄밀톤 키인즈, 위탄 게이트 웨스트, 어퍼 4th 스트리트 270에 위치한 MSU(주)로부터 구입할 수 있다. 상기 A/V 제어기/코프로세서(32)는 프로세서 인터페이스(60). 프로세서 캐쉬(62), 메모리 인터페이스/리프레쉬(64), 비디오 제어기(66), 인터럽트 제어기(68), 비디오 블리터(70), 선택적인 CD 블록 디코더, 디지털 신호 처리기(DSP)(74), 및 DSP 메모리(76)을 구비한다. 상기 프로세서 인터페이스(60), 메모리 인터페이스/리프레쉬(64) 및 비디오 제어기(66)는 집단적으로 비디오/메모리 제어기(67)로써 불리운다. 시스템 메모리(33), 중앙 프로세싱 유니트(30) 및 다른 장치는 A/V 제어기/코프로세서(32) 외부에 놓인다.

A/V 제어기/코프로세서(32)는 SYSTEM 버스(31)로부터 SYSTEM'버스(34)를 발생시킴으로써, 시스템 메모리(33)로부터 CPU(30)를 격리시킨다. 그래서 SYSTEM'버스(34)는 여러 가지 장치를 시스템 메모리(33)에 접

속한다. SYSTEM'버스(34)를 공유하는 것은 메모리 리프레쉬(64), 비디오 제어기(66), 선택적인 CD 블록 디코더(도시되어 있지 않음), DSP(74), 블리터(70), 및 CPU(30 : 프로세서 인터페이스 60을 통과하는)와 같은 각각 최상위 우선 순위에서 최하위 우선 순서로 정렬된 여섯 개의 가능한 버스 마스터이다. 버스 마스터중 단지 한 개만이 임의의 시간에 SYSTEM' 버스(34)를 제어할 수 있다. 비디오/메모리 제어기(67)내의 중재기는 이하에 서술되는 바와 같이 장치의 변하는 우선순위를 제어하고, A/V 제어기/코프로세서(32)내의 모든 장치와 전기 회선으로 통신한다. 예를 들면, CPU(30)은 인터럽트가 발생할 때까지 모든 버스 마스터의 가장 낮은 우선순위를 갖는다. 그래서, 중재기는 CPU 인터페이스(60)와 인터럽트 제어기(68) 모두와 회선으로 통신한다.

캐쉬(62)는 CPU(30)을 위한 명령을 프리훼치한다는 의미에서 캐쉬가 아니다. 그 보다는, 상기 캐쉬(62)는 변수, 스택 또는 프로그램 실행을 가속시키는 프로그램 코드를 위해 CPU(30)에 의해 사용될 수 있는 F14000H 내지 F143FFH에 위치된 512×16 비트 스태틱 RAM이다.

비디오/메모리 제어기(67)(프로세서 인터페이스 : 60, 메모리 인터페이스/리프레쉬 : 64, 및 비디오 제어기 : 66)은 SYSTEM' 버스(34)를 제어하고, 기술분야의 공지된 바와 같이 SYSTEM' 버스(34)에 부착된 메모리 장치를 위한 메모리 타이밍 신호(예를 들면, CAS,RAS,기록 인에이블 등)을 공급한다. 이것은 비디오 라인을 통하여 임의의 비디오 디스플레이 데이터를 훼치하고, 동적 RAM(DRAM)을 리프레쉬하기 위해 짧은 주기 동안 버스 마스터 동작을 보류시킨다. 이것은 또한 CPU(30)과의 인터페이스를 제어한다.

비디오 제어기(66)은 서로 다른 TV 표준에 적응되도록 프로그램될 수 있는 유연한(flexible) 비디오 타이밍 발생길르 구비하고, 최대 640×480 VGA 표준을 모니터 한다. 정확한 비디오 포맷은 A/V 제어기/코프로세서에 수평 주기, 수평 동기, 수평 블랭킹 종료, 수평 블랭킹 시작, 수평 디스플레이 시작, 수평 디스플리에 종료, 수평 훼치 시작, 수평 훼치 종료, 수평 수직 동기, 수직 주기, 수직 동기, 수직 블랭키 종료, 수직 블랭킹 시작, 수직 디스플레이 시작, 수직 디스플레이 종료, 비디오 인터럽트, 및 라이트 펜과 같은 다양한 레지스터를 세트함으로써 제어될 수 있다. 상기 비디오 제어기(66)은 이용가능한 세 가지 컬러 해상도 : 화소당 4비트, 화소당 8비트, 및 화소당 16비트를 갖는다. 스크린의 메모리 맵은 비디오 디스플레이 폭에 구애되어 있지 않지만, 독립적으로 정의 되어 있다.

비디오/메모리 제어기(67)은 80376 CPU(30)의 16 메가바이트 어드레스 범위를 시스템 RAM(000000H - 0FFFFFH)을 위한 1MB, 제 1 카트리지 ROM(400000 - 7FFFFFH)을 위한 4MB, 제 2 카트리지 ROM(800000 - BFFFFFH)을 위한 4MB, 오디오/비디오 제어기/코프로세서(F10000H - F1FFFFH)을 위한 내부 메모리 64 KB 및 시스템 ROM(FE0000H - FFFFFFH)의 128KB 블록과 같은 메모리 맵으로 디코드 한다. 내부 메모리인 상기 64 킬로바이트는 팔레트(palette) RAM, 블리터 레지스터, 및 DSP 레지스터 및 메모리를 구비한다. 팔레트 어드레스 범위는 상술하였다. 블리터 레지스터는 범위 F10400H에서 F107FFH까지 이른다. 상기 DSP 메모리는 F10800H에서 F1800H까지이다.

만약 선택적인 CD 드라이브가 시스템에 부착되어 있다면, 다음 영역이 메모리 맵에 부가된다 : 시스템 RAM(100000H - 1FFFFFH)의 다른 1MB와 CD 드라이브 (FC0000H - FDFFFFH)에 대해 128KB.

상기 인터럽트 제어기(68)은 CPU(30)에 대해 6개의 내부 인터럽트 : 비디오 인터럽트(최고 우선 순위), 아날로그 인터럽트 0(AI0), 아날로그 인터럽트 1(AI1), 아날로그 인터럽트 2(AI2), CD 블록 디코더 인터럽트, 및 DSP 인터럽트(최하위 우선순위)를 인터페이스 한다.

인터럽트 제어기는 CPU(30)이 인터럽트 승인 사이클 (cycle)을 수행할 때 인터럽트를 자동적으로 클리어(clear)한다. 마스크 비트는 각각의 인터럽트에 유용하다.

블릿터(blitter)(70)은 CPU(30) 또는 DSP(74)를 위한 하드웨어 그래픽 서브루틴으로서 동작하는 빠른 스크린 업데이트 및 애니매이션(animation)을 하기 위한 그래픽 프로세서이다. 이것은 CPU(30) 및 DSP(74)에 의해 메모리내로 기입된 명령어들을 실행한다. 이것은 새로운 명령어 세트를 시스템 메모리(33)으로부터 판독함으로써 긴 시퀀스의 그래픽 동작을 임의적으로 수행할 수 있다. 이것은 블릿터 프로그램 동작을 통한 버스 마스터로 되고 따라서 상당한 주기 동안에 SYSTEM' 버스(34)의 배타적 제어를 할 수 있다. 그러나, CPU(30)에 대한 그것의 우선순위는 절대적인 것이 아니고, 인터럽트가 발생할 때 CPU(30)에 SYSTEM' 버스(34)를 포기하는 것이 요구될 수 있다. CPU(30)은 시스템 레벨에서 가장 낮은 우선순위의 버스 마스터이다. 그러나 다른 하드웨어의 완전한 제어를 하고, 따라서, SYSTEM'버스(34)의 사용은 전적으로 CPU(30)의 프로그램 제어하에 있다.

블릿터(70)은 인텔리전트 블릿팅 동작을 가능하게 하는 만능 비교기(versatile comparator)와, 출력 데이터를 발생시키기 위한 논리 함수 유니트(LFU)를 갖는다. 논리 함수 유니트는 출력 데이타를 발생시키기 위한 여러 가지 유용한 방식으로 데이터 레지스터의 내용들을 조합시킬 수 있고, 비교기는 기입동작을 금지하고 블릿터 동작을 선택적으로 중지시키기 위해 데이터에 대한 소정의 비교를 수행할 수 있다.

논리 함수 유니트는 시스템 메모리(33) 내의 데스티네이션(destination)에 기입되는 출력 데이터를 발생시킨다. 이것은 소스와 데스티네이션 레지스터 화소들의 임의의 논리적 조합을 수행할 수 있다. 소스 데이터 화소는 소스 데이터 레지스터 또는 데이터 패턴 데이터 레지스터 중의 어느 하나로부터 선택될 수 있다. LFU는 데이터 레지스터로부터 2세트의 입력 데이터인 4개의 불 민텀(Boolean minterms : A&B, A&B, A&B, 및 A&B) 중의 하나를 선택하고, 2개의 선택된 민텀의 논리적 OR을 발생시킨다. 이것은 입력 데이터의 임의의 논리적 조합을 생기게 하여, 16개의 함수적 가능성이 존재한다.)

비교기는 소스, 데스티네이션, 및 패턴 데이터 레지스터내의 데이터에 대한 다양한 비교를 수행할 수 있다. 그 비교 조건이 맞으면, 이것은 금지 신호를 발생시킨다. 금지 신호는 기입 동작을 금지하고, 선택적으로 블릿팅 동작을 중지시키는데 사용된다. 비교기는 또한 충동 검출(collision detection) 및 시스템 메모리(33) 서치 동작과 캐릭터 페인팅에 도움주는 것과 같은 동작을 위해, 투명 색을 제공하도록 화소 플레인 효과를 제공하는데 사용될 수 있다.

DSP(74)는 음성 합성용의 간단하고 매우 고속인 프로세서로, 초당 명령어 (MIPs)를 3300만개 까지 처리할 수 있다. 이것은 DSP DMA 제어기(도시 안됨)을 경유하여 SYSTEM' 버스(34)에 억세스하여, 시스템 메모리

(33)에 바이트 또는 워드를 판독 및 기입할 수 있다. 이러한 전달은 단번에 발생하고 DSP 프로그램 제어하에 있게 된다. DSP(74)는 실질적으로 프로그램을 실행하고 그 자신의 전용 고속메모리(76)내에 데이터를 저장한다.

DSP(74) 오디오 코프로세서는 고성능 음악 합성기를 구현하기에 충분한 능력을 갖춘 범용 산술 코프로세서이다. 동기 직렬 출력은 16비트 정밀도를 가진 스테레오 오디오 신호의 발생을 위해 제공되고, 음성 질은 보통 컴팩트 디스크 기술과 관련된다. DSP(74)는 호스트 CPU(30)으로부터 마이크로-프로그램가능하고, 명령어 세트는 사용자가 음악 합성기의 기능과 다른 여러 가지의 다른 기능들을 수행하기 위해 장치를 프로그램할 수 있도록 충분히 신축성이 있다. 이러한 응용들은 알고리즘적 음성 발생, 패스트 푸리에 변환 기법을 이용하는 오디오 분석, 및 3차원 그래픽로테이션을 포함할 수 있다. DSP(74)는 최대 데이터 효율을 위해 하바드 아키텍처(Harvard architecture : 분리된 프로그램 및 데이터버스)를 사용한다. DSP(74)는 가산, 감산, 및 논리함수 뿐만 아니라 하드웨어 16비트×16비트 승산/누산을 수행하는 산술 논리 유니트(ALU)를 갖는다. 또한 틱당(tick) 한 개의 몫 비트를 발생시키는 별도의 직렬 제산 유니트가 있다.

DSP(74) 내의 ALU는 16비트 산술 논리 유니트이고, 이것은 본 기술 분야에 공지된 텍사스 인스트루먼츠 74181과 같은 기능을 가진다. 통상의 산술적 연산은 명령어로서 인코드된다 ; 통상적이지 않은 명령어는 범용 산술 명령어(GAI)로 ALU모드 비트를 직접 세팅함으로써 수행될 수 있다.

DSP(74)는 이에 연관된 DSP 메모리(76)을 갖는다. DSP메모리(76)은 프로그램 RAM, 데이터 RAM, 레지스터/상수 테이블 및 사인 ROM(모두 도시 안됨)을 포함한다. 일반적으로는 DSP(76)은 시스템 메모리(33)의 어드레스 공간 뿐만 아니라 DSP의 내부 어드레스 공간 모두에 억세스 가능하다. DSP 프로그램 RAM은 512개의 18비트 워드이다. 이러한 로케이션은 CPU(30)에 의해서만 기입될 수 있고, DSP(74)에 관한 한 판독 전용 프로그램(progRAM read-only)이다. 프로그램 RAM은 DSP의 내부 어드레스 공간에 나타나지 않는다. 프로그램 RAM은 DSP(74)가 실행하고 있을 때 호스트에 억세스할 수 없지만, DSP가 실행중지 상태일 때 억세스 할 수 없다.

DSP(74)는 또한 직렬 오디오 디지털-아날로그 변환기(DAC) 인터페이스를 갖는다. 직렬 DAC 인터페이스는 DSP(74)로 하여금 동기 직렬(I^2S 또는 유사) DAC를 구동시키고, CD 드라이브와 같은 동기 직렬 데이터 소스로부터 데이터를 입력하게 할 수 있다.

A/V 제어기/코프로세서(32)의 비디오 제어기(66)은 본 기술 분야에 공지된 바와 같이 비디오 제어기(66)으로부터의 18비트의 화소 정보(78) (적색, 녹색, 및 청색 각각 6비트)를 RGB 신호(80)으로 변환시키는 외부 비디오 DAC(50)에 접속된다. 비디오 DAC(50)의 각각의 색 채널[(R(80a), G(80b), 및 B(80c)]는 제1c도에 도시한 바와 같은 R2R 저항 트리와 2N2222 트랜지스터로 구현된다. 제1c도 내의 장치들은 도시한 바와 같은 회선으로 통신을 한다. 제1c도 내의 저항 (86a-86j)는 5% 오차 허용도 내에서 도시된 저항값을 갖는 모두 0.25와트 저항이다. 트랜지스터(88)은 2N2222이다.

제1b도를 다시 참조하면, RGB 신호(80)은 NTSC/PAL 인코더(52)에 의해 NTSC 합성 비디오 신호(90)으로 변환된다. NTSC/PAL 인코더(52)는 A/V 제어기 / 코프로세서(32)의 비디오 제어기(66)에 의해 발생된 크로마 클럭(92), HSYNC 및 VSYNG 신호(94) 및 비디오 DAC(50)에 의해 발생된 적생(80a), 녹색(80b) 및 청색(80c) 비디오 출력을 받아들여, 공지된 NTSC 또는 기저대역(baseband) 비디오 포맷으로 합성 비디오 신호(90)을 발생시킨다. 대안으로서, 공지된 PAL(유럽 텔레비전 신호기준) 포맷이 발생될 수 있다. 합성 비디오 신호(90)은, 본 기술 분야에 공지된 바와 같이, 단일한 암형(female) RCA 포노 잭(도시안됨)으로 외부 장치들과 접속된다. 보다 바람직한 실시예에서, NTSC/PAL 인코더(52)는 소니사(Sony Corp.)에서 제소한 CXA1145이다. 대안으로, 모토롤라사(Motorola Corp.)에서 제조한 MC1377이 사용될 수 있다.

오디오 ADC / DAC / CODEC(54)는 공지된 필립스 I^2S 프로토콜에 따르는 직렬링크(96)으로 DSP(74)에 링크된다. ADC / DAC / CODEC(54)는 아날로그 데이터를 디지털 데이터로 변환시키고, 또 그 반대로도 변환시키며, 디지털 데이터를 컴프레스(compress)하고 디컴프레스(decompress)한다. ADC / DAC / CODEC(54)는 선택 마이크로폰으로부터의 외부 스테레오 아날로그 데이터(97a-97b)를 A/V 제어기/코프로세서(32)에 인터페이스시킨다. 오디오 입력(97a-97b)는 표준 스테레오 1/4접속기로 외부 장치들과 접속된다. 오디오 ADC / DAC / CODEC(54)는 좌우 오디오 라인 아웃 신호(98a-98b)를 발생시킴으로써 A/V 제어기/코프로세서로부터의 디지털 데이터를 외부 장치들에 인터페이스시킨다. 이들 신호(98a-98b)는 본 기술 분야에 공지된 바와 같이, 선택 스피커(도시안됨)와 같은 외부 장치들에 2개의 암형 RCA 폰 잭으로 접속된다. 아래에 설명되는 바와 같이, 오디오 라인 신호(98a-98b)는 또한 RF 비디오 신호(22)에 가산된다.

보다 바람직한 실시예에서, ADC / DAC / CODEC(54)는 크리스탈 세미콘덕터사(Crystal Semiconductor)에서 CS4216이다. 이 부품은 프로그래머블 감쇠기를 갖는 출력뿐만 아니라, 프로그래머블 이득을 갖는 마이크로폰 입력을 포함한다. 이득 및 감쇠는 모두 DSP(74)에 의해 프로그램 가능하게 제어된다.

다른 방식으로서, ADC / DAC / CODEC(54)는 필립스사(Philips)에서 제조하는 TDA 1311 DAC로 대체될 수 있다. 이러한 칩이 사용되면, ADC 및 CODEC 기능 들이 쓰이지 않을 것이다.

RF 변조기(56)은 TV(16)에 직접 입력되기에 적합한 RF 비디오 신호(22)를 발생시키기 위한 캐리어 주파수상에서 오디오 ADC / DAC / CODEC(54)로 부터의 좌우 오디오 라인 아웃 신호(98a 및 98b)와 NTSC/PAL 인코더(52)로부터의 합성 비디오 신호(90)을 병합시킨다. 다른 PAL(유럽 텔레비젼 신호 표준) 및 NTSC 포맷을 발생시키기 위해 다른 RF 변조기 및 수정(crystal)이 사용되어야 한다. RF 비디오 신호(22)는 본 기술 분야에 공지된 바와 같이 단일 한 암형 F 동축 접속기로 외부장치들과 접속된다.

제2a도-제2m도 및 제3도를 참조하면, 본 발명의 입력 장치(18)의 실시예가 도시된다. 이 도면에 도신된 바와 같이, 입력 장치(18)은 터치패드(19)와 단일 인클로져(100) 내에 인클로져된 2개의 조이스틱(20a, 20b)를 포함한다. 인클로져(100)은 ABS-T (물질 ABS, 등급 T로서 와이 입 스트리트 180, 운 통, 코우룬, 홍콩에 있는 윙스 인더스트리얼 센터로부터 이용할 수 있는 아크릴로니트릴레-부타디엔-스티렌)으로 만들어진다. 입력 장치(18)은 템플리트 오버레이(102)를 받아들이고, 오버레이 각각은 몸체(104), 탭(106) 및 바닥의 옛지에 있는 식별 패턴(108)을 포함한다. 템플리트 오버레이(102)는 코팅된 페이퍼, 코팅된 카드

보드, 또는 폴리에스터 필름과 같은 얇은 물질로 만들어진다. 한가지 적합한 폴리에스터 필름은 듀퐁사에 의해서 제조된 것이고 듀퐁의 Mylar 상표로 시판되고 있다. 몸체(104)는 그 안에 또는 그위에 인스크라이브된(inscribed) 그래픽 영상을 갖는다. 즉, 그래픽 디자인이 기입되고, 프린트되고, 페인트되고, 조각되고, 새겨지고, 실크 스크린되거나 또는 다른 경우에는 오버레이 몸체(104) 내부 또는 그 위에 영구적으로 부착된다. 탭(106)은 몸체(104)로부터 연장되고 템플리트 오버레이(102)를 죄는데 사용된다. 식별 패턴(108)은 제2g도-제2k도를 첨부한 텍스트에서 설명될 것이다.

터치패드(19)는 손가락, 철필(21) 또는 이와 유사한 것과의 접촉이 가능하도록 노출된 패드 표면(110)을 갖는다. 패드 표면은 패드 표면(110) 내에 또는 위에 인스크라이브된 디폴트 템플리트 그래픽 디자인을 갖는다. 즉, 디폴트 템플리트 그래픽 영상은 영구적으로 기입, 프린트, 페인트, 조각, 새김, 실크 스크린되거나 또는 다른 경우에 패드 표면(110)상에 또는 위에 고정된다.

제2a도에 도시된 바와 같이 엔터(enter), 엑싯(exit), 프리비어스(previous), 넥스트(next) 및 화살표 키(위, 아래, 좌, 우 : up, down, left 및 right)와 같은 기능들이 패드 표면(110)내에 또는 위에 인스크라이브된 디폴트 템플리트 그래픽 디자인에 의해 지원된다. 대안으로서, 셀렉트(select)가 엔터 대신에 사용될 수 있고, 캔슬(cancel)이 엑싯 대신에 사용될 수 있다. 또 다른 대안으로서, 10개의 직사각형 영역이 정해질 수 있는데, 하나 하나가 각 아라비아 숫자에 대응한다. 또 다른 대안으로서, 영어의 문자가 각각 터치패드의 영역에 매핑될 수 있다. 또 다른 대안으로 QWERTY 키보드가 패드 표면(110)내에 매핑될 수 있다. 실제적으로, 가상적인 어떤 패턴 또는 패턴과 기호의 조합이 선택될 수 있다. 디폴트 그래픽 영상은 시스템(10)을 위해 목표된 여러 가지의 응용에 유용하도록 선택되어야 한다.

입력 장치(18)은 패드 표면(110)에 인접하여 템플리트 오버레이(102)를 고정시키기 위한 2가지 형태의 리테이너(retainer)를 포함하는데 다음과 같다 ; (1) 일반적인 U자형 슬롯(114)를 정의하고 템플리트 오버레이(102)를 세 측면상에 리테인하는 3개의 직선 모서리(straight lip)을 포함하는, 오버행임 모서리(overhanging lip : 112) (2) 네 번째 측면상에서 템플리트 오버레이(112)를 리테인하는 한 쌍의 릿지(ridges:116a,116b). 모서리(112), 슬롯(114) 및 릿지(116a, 116b)는 이후 보다 상세히 도시되고, 제2c도, 제2d도 및 제2e도에 관련한 텍스트에서도 설명될 것이다. 제2a도에는 또한 입력 장치(18)을 휴대하기 위한 핸들(118)과 철필(21)을 보관하기 위한 튜브 형태의 애퍼추어(tubular aperture:120)이 도시되어 있다.

제2b도를 참조하면 패드 표면(110)에 인접한 위치 내에 템플리트 오버레이(102)가 있는 입력 장치(18)의 평면도가 도시된다. 이 도면 도시된 바와 같이, 오버레이(102)의 몸체(104)의 세 측면은 세 모서리부(112a-112c) 하에서 슬립(slip)된다. 이 도면에는 또한 오버레이(102)의 탭(106)의 양 측면 상에 하나씩 위치되어 있는 오버레이(102)의 몸체(104)의 네 번째 측면을 리테인하는 릿지(116a, 116b)가 도시되어 있다. 오버레이(102)는 모서리 부분(112a 및 112c) 하에서 오버레이(102)의 몸체 (104)의 좌우측면을 슬립시켜 삽입시키고, 오버레이(102)의 몸체(104)의 바닥 엣지가 다른 모서리 부분(112b)의 아래에 있을 때까지 오버레이(102)를 아래로 하여 슬라이딩시킴으로써 삽입된다. 최종적으로 오버레이(102)는 해제되고 탭(106)은 상부에서 오버레이(102)를 리테인하는 릿지(116a, 116b) 사이에 네스트(nest)된다.

제2c도, 제2d도 및 제2e도는 릿지(116a, 116b) 및 오버레이(102)를 상세하게 도시한 것이다. 제2d도는 릿지(116b)를 넘어서 연장한 탭(106)을 도시한다. 제2e도는 릿지(116b)에 인접한 오버레이(102)의 몸체(104)를 도시한다.

오버레이(102)는 탭(106)을 엄지 손가락과 집게 손가락으로 잡고 오버레이(102)의 몸체(104)를 릿지(116a, 116b) 위로 올림으로써 제거되어 오버레이(102)가 U자형 모서리(112) 아래로부터 밖으로 그리고 릿지(116a, 116b)위로 미끄러지게 된다.

제2f도는 3개의 측면 상에 오버레이(102)를 리테인 하는 슬롯(114)를 도시한다. 이 도면에는 또한 터치패드 센서(122), 오버레이 센서(124), 다수의 템플리트 오버레이(102)를 저장하기 위한 캐비티(126) 및 센서(122)의 접촉이 감지될 수 있기에 충분한 저항력을 제공하는 인클로져(100)과 동일한 물질로 이루어진 실질적으로 고정된 베이스(127)이 도시되어 있다.

터치패드 센서(122)는 상기 패드 표면(110)에 인접하여 위치되어 있고 손가락, 철필(21) 또는 그와 유사한 것에 의해 패드 표면(110)상에 또는 그에 인접하는 압력이 센서(122)로 하여금 접촉된 위치를 검출할 수 있도록 구성되어 있다.

터치패드 센서(122)는 임피던스에 기초한 센서, 음향센서 및 스위치 폐쇄형 센서와 같은 여러 형태중 임의의 한 형태일 수 있다. 예로서는 멤브레인 스위치 매트릭스 (membrane switch matrixes)가 있는데, 이는 발명자 플로렐라(Florella)에게 허여된 미합중국 특허 제 4,736,190호 및 발명자 이토(Ito)등에 허여된 미합중국 특허 제4,529,959호에 개시되어 있는 장치일 수 있다. 적당한 센서(102)는 부품번호 PR39983로서 홍콩 코울룬 크운 통 와이 입 스트리트 180 윙 인더스트리얼 센터 소재의 윙 전자 주식회사에서 구매할 수 있다.

센서(122)는 패드 표면(110)을 접촉하거나 패드 표면(110)에 인접한 템플리트 오버레이(102)를 접촉하는 철필(21), 손가락 등에 반응하여 적어도 하나의 전기신호를 발생한다. 예를 들면, 스위치 폐쇄형 센서들은 한 축상에 있는 많은 라인들을 순차적으로 구동하기 위한 다수의 구동기와 다른 축상에 있는 라인들중 어느 라인이 구동기에 의해 구동된 신호를 전달하는지를 검출하기 위한 다수의 수신기를 통상 필요로 한다. 어느 수신기에 의해 검출된 신호를 어느 구동기가 발생했는가를 알면 스위치 폐쇄를 유도하는 접촉의 위치를 결정할 수 있다.

오버레이 센서(124)는 오버레이(102)상의 식별 패턴(108)에 해당되는 전기 신호를 발생하는 센서이다. 그래서, 터치패드(19)의 오버레이 센서(124)와 템플리트 오버레이(102)의 식별 패턴(108)은 위치 및 장치 기술에 있어서 둘다 일치해야만 한다. 적당한 오버레이 센서(124)는 앵글형 송/수신기를 각각 갖고 있는 일렬로 된 6개의 광전 송/수신기이다. 이들은 본기술 분야에서 공지되어 있으며 부품번호 PR 39990으로서 윙 전자 주식회사에서 구입할 수 있다.

제2g도에 도시되어 있듯이, 오버레이 센서(124)는 인클로저(100)내의 6개의 구멍(128a-128f)을 통해서 식별 패턴(108)에 광학적으로 결합된다. 각각의 구멍(128)아래에는 하나의 광선 송/수신기 쌍(도시안됨)이 있다. 다른 실시예에서(도시 안됨), 오버레이 센서(124)의 6개의 송/수신기 쌍 및 6개의 구멍(128a-128f)는 3개씩 두 그룹으로 나뉘어져서 터치패드 센서(122)의 양쪽에 하나씩 배치된다. 즉 3개의 구멍(128a-128c : 및 그들의 관련된 송/수신기 쌍)은 하나의 모서리 부분(lip portion : 112a) 아래에 배치될 수 있고, 다른 3개의 구멍(128d-128f : 및 그들의 관련 송/수신기 쌍들)은 다른 모서리 부분(112c)아래에 배치될 수 있다.

제2h도-제2k도는 식별 패턴(108)의 예들을 보여준다. 식별 패턴(108)은 템플리트 오버레이(102)의 한 엣지상에 배치된다. 오버레이 센서(124)의 각각의 개별적인 송/수신기 쌍에 대해 6개의 개별적인 마크가 있다. 오버레이(102)가 제 위치에 있을 때, 식별 패턴(108)은 센서(124)와 정열된다. 대안적인 실시예에서, 3개의 구멍들로 이루어진 두 그룹이 센서용으로 사용된다면, 식별 패턴(108)은 같은식으로 두 그룹으로 나뉘어져서 터치패드 센서(122) 양쪽에 하나씩 배치되어야 한다.

센서(124)로서 광전 송/수신기 쌍들을 이용하면 템플리트 오버레이(102)내에 아주 간단한 식별 패턴을 설계할 수 있다. 오버레이를 만드는 재료가 흰색이면, 오버레이 뒤쪽에 가해지는 흑색 잉크 또는 페인트 영역들이 한 형태의 식별 패턴으로 사용되고 흑색 잉크가 없는 흰색 영역들은 다른 식별 패턴으로 사용될 수 있다.

식별 패턴에 대한 여러 가능한 조합들의 예가 제2h도 - 제2k도에 도시되어 있다. 재2h도 내지 제2k도는 2진 패턴 010001_2,011110_2,000000_2 및 111111_2 각각에 대응하는 식별 패턴(108)을 보여주고 있다. 이러한 식별 패턴은 템플리트 오버레이 (102)의 엣지를 따라서 간격을 둔 밝고 어두운 영역의 그룹이 나타낸다. 000000_2 패턴은 어떤 탬플리트 오버레이(102)도 존재하지 않음을 나타내므로 아마도 이용되지 않을 것이다. 이 경우에는 디폴트 템플리트 그래픽 이미지가 사용될 것이다.

본 발명의 입력 장치(18)의 정면도인 제21도는 터치패드(19)의 각 사이드에 하나씩 있는 조이스틱(20a, 20b)의 위치를 보여준다. 이 도면에 도시되어 있듯이, 조이스틱(20a,20b) 각각은 그들 각 단부에 고정되어 있는 순간적인 푸쉬버튼 스위치(130a, 130b)를 갖고 있다. 제2I도는 또한 다수의 템플리트 오버레이를 수용하는데 이용되는 캐비티(126)로 통하는 직사각형 애퍼추어(132)를 도시하고 있다.

제2a도의 선 2M-2M으로 지정된 평면을 따라 본질적으로 취해진 부분 단면도인 제2m도는 본 발명의 입력 장치에 사용되는 조이스틱에 대한 세부도이다. 제2m도의 세부사항은 두 개의 조이스틱(20a, 20b)에 대하여 반복된다. 게다가, 제2m도의 세부 사항은 90도 회전할 때 대칭이다. 그래서 많은 구조 중 2개의 구조가 이 도면에 도시되어 있을지라도, 실제로는 동일한 4개의 구조가 이 실시예에 이용된다. 조이스틱(20a)은 로드(134)에 고착되어 있고, 이 로든 애퍼추어(136)을 통해서 인클로져(100)에 의해 정의된 볼륨내로 연장된다. 로드(134)는 피봇 포인트(138)에서 종료된다. 이 피봇은 스위치 베이스(140)상에서 피봇 운동을 한다. 스위치 베이스(140)은 물리적으로 인클로져(100)에 부속되어 있는 몇몇 스탠드오프(144a-144d)내로 들어가는 4개의 나사(142a-142d)에 의해 인클로져(100)에 고착된다. 애퍼추어(136)은 환형 실(annular seal : 146)에 의해 밀봉되고, 이 실의 환형은 로드(134) 둘레에 꼭 맞게 되어 있고 그의 외부 엣지는 통상의 방법에 의해서 애퍼추어에서 인클로져(100)에 물리적으로 부속되어 있다.

조이스틱(20a)은 피봇 포인트 14도 만큼 자유로이 움직일 수 있다. 즉, 로드(134)는 실제로 모든 방향으로 스위치 베이스(140)에 의해 정의된 평면에 대해 완전한 수직을 기준해서 7도 움직일 수 있다. 로드(134)는 자신에 물리적으로 부속된 4개의 스위치 작동기 암(actuator arm : 150a-150d)을 갖고 있다. 작동기 암(148)은 스위치 베이스(140)에 물리적으로 부속되어 있는 4개의 고무로 된 돔형 순간 푸쉬버튼 스위치(150a-150b)에 인접하게 배치되어 있다. 암(148) 및 스위치(150)은 조이스틱(20a)가 수직으로부터 변위될 때, 하나 이상의 암(148)이 그들과 관련된 스위치(150)을 폐쇄하도록 구성되어 있다. 그래서, 스위치(150)가 폐쇄되면 조이스틱(20a)가 움직였음을 알 수 있게 된다. 대안적으로, 조이스틱(20a, 20b)는 본 기술분야에 알려져 있는 전위차계 내장 시스템(potentiometer-based SYSTEMs)과 같은 다른 구조로 실현될 수도 있다.

제3도를 보면, 입력 장치(18)내의 회로에 대한 블록도가 도시되어 있다. 입력 장치(18)은 터치패드 센서(122), 오버레이 센서(124), 조이스틱 센서(200a, 200b), 좌표 결정 회로(202), 오버레이 검출 회로(204), 100밀리세컨드 타이머(208) 및 인터페이스 회로(210)을 포함하고 있으며, 이들 모두는 제3도에 도시된 바와 같이 전기회전으로 통신을 할 수 있게 연결되어 잇다.

좌표 센서(122) 및 오버레이 센서(124)는 제2도를 참조로 앞서 기술한 바와 같다. 좌표 결정 회로(202)는 좌표 센서(122), 인터페이스 회로(210) 및 100 밀리세컨드 타이머(208)와 회선으로 통신을 한다. 좌표 결정 회로(202)는 좌표 센서(122)로부터의 전기 신호(203)을 받아들인 다음 손가락, 철필(21)등에 의해 접촉된 위치에 해당하는 X축 및 Y축 값을 결정한다. 예를 들어, 좌표 센서(122)가 스위치형 센서인 경우, 좌표 결정 회로(202)는 이 기술분야에 공지되어 있는 바와 같은, 어느 스위치가 폐쇄되었나를 결정하기 위한 구동기 및 수신기를 구비하고 또한 상기 스위치의 위치를 패드 표면(110)에 대한 의미 있는 값으로 번역하기 위한 로직을 구비할 것이다.

오버레이 검출 회로(204)는 오버레이 센서(124) 및 인터페이스 회로(210)와 회선으로 통신을 한다. 오버레이 검출 회로(204)는 오버레이 센서(124)로부터의 전기 신호(205)를 수신하여, 식별 패턴(108)에 해당하는 메시지를 발생한다. 오버레이 센서(124)로부터 전기신호가 없는 경우에는 상술한 바와 같이 000000_2가 검출된다.

방향 결정 회로(206)은 조이스틱 센서(200a 및 200b) 및 인터페이스 회로(210)과 회선으로 통신을 한다. 조이스틱 센서(200a 및 200b)는 상술한 바와 같이 4개의 고무로된 돔형 스위치(150a - 150d)와 두 개의 조이스틱 스위치(130a, 130b)를 포함한다. 방향 결정 회로는 이를 스위치들이 폐쇄된 것에 근거해서 메시지를 발생한다. 100 밀리세컨드 타이머(208)은 좌표 결정 회로(202)와 회선으로 통신을 한다.

타이머(208)은 100 밀리세컨드 기간의 만료를 반복적으로 결정하여 이 기간의 만료를 가리키는 신호(20

9)를 발생한다. 좌표 결정 회로(202)는 타이머(208)에 의해 검출된 100 밀리세컨드 기간들의 만료 동작 사이에서 손가락, 철필(21) 등에 의한 접촉 위치 변화를 검출하는데, 신호(209)를 이용한다.

인터페이스 회로(210)은 좌표 결정 회로(202), 오버레이 검출 회로(204), 방향 결정 회로(206), 및 (직렬 데이터 라인(22)를 통해서) 데이터 프로세싱 유니트(12)와 회선으로 통신하며, 직렬 데이터 라인 확장선(23)을 통해서 다른 입력 장치와도 회선으로 통신을 한다. 인터페이스 회로(210)은 좌표 결정 회로(202)에 의해 결정된 좌표값, 오버레이 검출 회로(204)에 의해서 발생된 오버레이 메시지 및 방향 결정회로(206)에 의해 발생된 메시지를 받아들인 다음, 직렬 데이터 링크(22)를 경유해서 그러한 정보를 데이터 프로세싱 유니트(12)에 전송한다.

모든 입력 장치들은 프로세싱 유니트(12)와 연쇄적으로 연결된다. 그래서, 인터페이스 회로는 다른 입력 장치들로부터의 패킷을 CPU(30)로 보내야만 한다. 이하 좀더 상세히 설명되겠지만, 프로세싱 유니트(12)에 연결된 입력 장치 각각은 고유한 장치 번호를 갖고 있다. 프로세싱 유니트(12)에 가장 가까운 장치는 장치번호 0을 갖고 있고, 프로세싱 유니트(12)로부터 더 멀리 떨어져 있을수록 더 큰 장치 번호를 갖게된다. 그러나, 입력장치들은 그들 자신 또는 다른 장치의 장치 번호는 알지 못한다. 그래서, 각각의 장치는 동일한 종류의 다른 입력 장치로부터 전달된 데이터 패킷의 장치 번호에 1을 더해야만 한다. 15 이상의 장치 번호와 연결되어 있는 임의의 입력 장치는 무시된다.

예를 들어, 동일한 종류의 3개의 입력 장치 α, β, γ는 다음과 같이 프로세싱 유니트(12)에 연결된다고 가정한다 : α는 프로세싱 유니트(12)에 연결되고, β는 α에 연결되고, γ는 β에 연결된다. 따라서, α는 장치 번호 0을 갖고, β는 장치 번호 1을 가지며, γ는 장치번호 2를 갖는다. 다른 장치들은 그들 자신 또는 다른 장치 번호를 알지 못한다. 각각의 입력 장치는 장치번호 0을 갖고 있는 자신의 데이터 패킷을 전송한다.

α가 프로세싱 유니트(12)에 데이터 패킷을 보낼 때, 디폴트 장치 번호 0은 α가 프로세싱 유니트(12)에 가장 근접해 있기 때문에 올바른 것이 된다. 그러나, β가 및 γ 또한 장치 번호 0을 갖고 있는 데이타 패킷을 전송한다. 이러한 상황을 치유하기 위해서, 각각의 장치는 전달되는 패킷의 장치 번호에 1을 더한다. 그래서, β 가 데이터 패킷을 γ로부터 α로 보낼 때, β는 1을 장치 번호에 부가시킨다. 그 결과, γ로 부터의 패킷에 장치 번호 1이 주어진다. 비슷한 식으로, α가 γ데이타 패킷을 프로세싱 유니트(12)로 보낼 때, α는 장치번호에 1을 더한다. 그 결과, γ로 부터의 패킷에는 올바른 장치 번호 2가 제공된다. 그래서, 체인식으로 연결된 각각의 장치는 다음 장치로 보내지는 동일한 형태의 장치들로 부터의 각각의 데이터 패킷의 장치 번호에 1을 더한다.

따라서, 다른 입력 장치(있다면)로부터 수신된 데이터 패킷들을 전달하는 것이외에도, 인터페이스 회로(210)은 직렬 데이터 라인 확장(23)을 경유해서 수신된 동일한 형태의 장치들로부터의 임의의 데이터 패킷내에 있는 장치 번호에 1을 더한다. 인터페이스 회로(210)은 수정된 장치 번호 및 수정되지 않은 장치 번호를 갖고 있는 데이터 패킷을 데이터 프로세싱 유니트(12)로 전달한다.

본 발명의 입력 장치(18)을 갖고 있는 시스템(10)을 이용하면 매우 자명해진다. 입력 장치들은 직렬 링크(22)를 통하여 데이터 패킷을 데이터 프로세싱 유니트(12)로 보낸다. 상술한 바와 같이, 입력 장치들은 I/O 코프로세서(36)를 통해서 CPU(30)와 인터페이스한다. 각각의 입력 장치는 다음의 입력 장치에 연쇄적으로 (daisy chained)연결된다. I/O 코프로세서(36)은 데이터 패킷을 수신하여 선입 선출(F1F0) 방식으로 그들을 저장한다.

매 50 msec 틱(tick)마다, I/O 코프로세서(36)은 CPU(30)을 인터럽트한다. 응답시에, CPU는 CPU에 의한 최종 억세스 이후로의 틱 수와 상술한 바와 같이 전송된 장치 메시지 번호를 결정하기 위하여 코프로세서(36)의 I/O 포트 AS0에 있는 단일 바이트를 억세스한다. 10 종류의 장치 메시지들이 다음 표에 도시되어 있다.

[표 1]

입력 장치 메시지 구조

장 치 유 형	0 바이트 (카운트)	1 바이트(장치 ID)		2 바이트	3 바이트	4 바이트
		4-7비트	0-3비트			
키보드 (PS/S)	2	0	0	스 캔 코 드	해 당 없 음	해 당 없 음
마우스 (PS/2)	4	1	0	마우스 바이트 1 (버 튼)	마우스 바이트 2 (X 데이타)	마우스 바이트 3 (Y 데이타)
스위치 닫 힘	변 수 1~255	체인 내 장 치 번 호	1	버튼 상태 0 : 열림 1 : 닫힘	버튼 상태 (옵 션)	버튼 상태 (옵 션)
조이스틱 (디지탈)	2	체인 내 장 치 번 호	2	스위치닫힘 / 개방코드	해 당 없 음	해 당 없 음
(상대) 좌 표	4	체인 내 장 치 번 호	3	버 튼 상 태	델타 X	델타 Y
(절대) 좌 표	4	체인 내 장 치 번 호	4	버 튼 상 태	X 좌표	Y 좌표
터치패드 오버레이 메 시 지	2	체인 내 장 치 번 호	5	오버레이 코 드 (0~63)	해 당 없 음	해 당 없 음
명 령 메시지	변 수 1~255	체인 내 장 치 번 호	6	명령버튼 데 이 타	(옵션)	(옵션)
시스템 패 스 스 루	변 수 1~255	체인 내 장 치 번 호	14	0 바이트	1 바이트 (옵 션)	2 바이트 (옵 션)
초기화 메세지	변 수 1~255	체인 내 장 치 번 호	15	장 치 타 입 (ID)	Mfg 코드 0 바이트	Mfg 코드 1 바이트 (옵 션)

표1에서 알 수 있는 바와 같이, 메시지 구조들은 서로 다른 길이를 갖고 있으며, 그들이 해당되는 입력 장치들에 밀접하게 관련된 구조를 갖고 있다. 표에 있는 장치 메시지들은 개개의 I/O 장치로부터 I/O 코프로세서에 보내진 데이터에 대한 것과 I/O 코프로세서에 의해 CPU로 보내진 데이터에 대한 메시지와 동일하다. 앞서 보여준 구조 이외에도, I/O 장치로부터 I/O 코프로세서로 가는 각각의 메시지가 입력 장치(18)로부터 정확한 데이터(uncorrupted DATA : 18)가 전송되는 것을 보장하기 위한 검사함(checksum)을 갖고 있다. 이 검사합은 표준 모듈로 256 검사합(standard modulo 256 checksum)으로, 검사합 값은 모든 바이트의 합이 0으로 되게 하는데 필요한(합산 동안 캐리(carry)는 무시함) 값이다. I/O 코프로세서는 데이터를 CPU에 전송하기 전에 검사합을 분해한다. 따라서, CPU에 의해 판독된 바이트의 스트림은 다음의 예외만 제외하고는 I/O 코프로세서에 의해 수신된 바이트의 스트림과 실질적으로 동일하다 : (1) CPU에 의해 판독된 제 1 바이트는 스틱 수 및 I/O 장치 메시지의 번호를 포함하는 특정 바이트인 경우 및 (2) 검사합이 유실(missing)되는 경우.

PS/2 마우스 및 키보드 장치들은 장치 형태 0으로서 지원된다. 키보드는 체인 번호 0을 갖고 있고, 마우스는 체인번호 1을 갖고 있다. 이들 장치들은 직렬 데이터 링크(22)를 통하여 현존한느 PS/2 프로토콜을 이용해서 I/O 코프로세서에 의해 지원된다.

이러한 장치 형태 1은 다중 버튼을 갖고 있는 장치로 지정되어 있다. 이러한 메시지 형태를 사용하면 시스템에 255 바이트(바이트당 8개의 버튼), 즉, 2040개까지의 버튼이 입력될 수 있다. 개방 버튼은 논리 ZERO으로서 전송되지만, 폐쇄 버튼은 논리 ONE으로서 접속된다. 이는 가변 길이 메시지이다.

조이스틱(20a, 20b)와 같은 디지털 조이스틱들은 장치 유형 2로서 지원된다. 두 개의 조이스틱은 각각의 터치패드(19)와 연관되어 있다. 각각의 조이스틱은 유일한 체인번호를 갖고 있다. 왼쪽에 있는 조이스틱 각각은 기수(1,3,5,7,9 등)이고, 오른쪽에 있는 조이스틱 각각은 우수(0,2,4,6 등)이다. 각각의 조이스틱은 독립적으로 보고된다. 메시지는 고정된 길이의 메시지이다. 디지털 조이스틱 센서들이 다수의 스위치(150a-150d)를 포함한다는 것을 상기하자. 메시지는 8개까지의 스위치들을 나타내는 바이트이다. 이들 스위치는 움직임 센서 스위치(150a-150d) 및 스위치 (130a)와 같은 데이터 입력 스위치를 포함한다. 이러한 형태의 메시지 바이트의 개개의 비트는 업 스위치(MSB), 다운 스위치, 레프트 스위치, 라이트 스위치, 스위치 #1, 스위치 #2, 스위치 #3, 및 스위치 #4 (LSB)를 나타낸다. 터치패드(19)를 포함하고 있는 조이스

틱(20a, 20b)은 하나의 버튼만(130)을 갖고 있으며, 이것은 상기의 스위치 #1에 대응한다. 다른 3개의 버튼은 항상 0으로서 보고된다.

마우스 및 트랙볼과 같은 좌표 장치들은 장치 형태 3으로서 보고된다. ID를 뒤따른 제1 바이트는 장치에 대한 버튼 정보를 보고하는 것이다. 최대 8개까지의 버튼이 보고될 수 있다. 그 다음 바이트는 델타 Y 값이 뒤따르는 델타 X값이다. 델타 X 및 델타 Y 값은 장치의 최종 보고된 위치에 근거한다. 응용 프로그램들은 필요하다면 이들 값을 절대 좌표로 변환해야만 한다. 최대 움직임은 255이다. 실제 움직임이 255를 초과하면, 2개 이상의 메시지가 전송될 것이다. 이것은 고정된 길이의 메시지이다.

터치패드(19)는 장치 형태 4로서 지원된다. 이러한 장치 형태의 다른 장치들은 아날로그 조이스틱을 포함한다. ID를 뒤따르는 제1 바이트는 버튼 정보를 보고하는데 이용된다. 그 다음의 바이트는 절대 X 위치를 보고하는데 이용된다. 절대 Y 위치는 그 다음 순서에 있게 된다. 절대 X 및 Y 값은 각각 1바이트이고 0-255 범위내에 한정된다. 이는 정해진 길이의 메시지이다.

터치패드 오버레이(102)는 장치 형태 5로서 보고된다. 터치패트 오버레이들은 터치패드 내의 6비트 센서(124)를 이용하여 감지된다. 오버레이 변화가 터치패드에 의해 감지되면, 메시지가 발생된다. 모든 오버레이 코드들은 응용 종속이며 응용 프로그램은 각 오버레이 코드들을 인식해야만 한다. 이 메시지는 정해진 길이의 메시지이다.

액션 메시지(action message)는 여러 방법으로 다중 장치 형태에 의해 발생 될 수 있는 미리 정의된 장치에 독립적인 기능의 한 세트를 정의하는데 이용되지만 같은 방법으로 시스템 및 응용 프로그램들에 의해 사용되고 해석된다. 액션 메시지는 가변 길이 메시지를 이용하는 장치 형태 6으로서 보고된다. 이 특정 실시예에서 3개의 장치-독립성 기능들이 정의되어, 이 바이트의 하위 3개의 비트와 관련되어 진다: 각각 3개의 장치-독립성 기능은 START(활동 또는 처리를 시작하다), PAUSE(활동 또는 처리를 중지하다), 및 SELECT(여러 사건 또는 액션들 중 하나를 선택하다). 비트들은 이들 기능을 보고하도록 설정되어 있다. 모든 다른 비트들은 미래의 사용을 위해 예비된 것이며, CPU에 0으로서 보고된다.

시스템 통과 메시지(pass-through message) 형태는 앞서 정의된 장치 형태에는 적용할 수 없는 장치 형태를 조정하는데 이용된다. 메시지 형태 14가 이용된다. 이것은 가변 길이 메시지이다. 데이터의 정의는 장치-종속성이고 특정 응용이다. 각각의 응용은 이런 형태의 메시지를 요구되는 기능으로 변환시켜야만 한다.

각 장치로부터의 제1 메시지는 장치 형태 15이다. 이것은 어떤 장치가 입력 메시지를 전송한다는 것을 시스템에 알리는데 이용된다. 이 메시지는 또한 입력을 보고하는데 사용될 미래의 장치 형태를 정의한다. 이것은 가변 길이 메시지이다.

시스템이 구동되면 I/O 코프로세서는 50 ms 간격으로 구성을 결정해서 시스템에게 알리기 위해서 카트리지 및 확장 감지 라인을 주사해서(scan) 구성 바이트(configuration byte)를 CPU에 전송한다. 이것은 구동될 때 CPU가 I/O 코프로세서로부터 수신하는 제1 바이트이다. I/O 코프로세서는 변화가 감지될 때 모듈 구성 인터럽트를 발생한다; 카트리지 상태의 변화는 시스템을 리세트되게 한다. 결과 I/O 코프로세서가 다른 구성 바이트를 CPU에 전송하게 된다. 전송된 바이트 내의 적절한 비트 세트는 관련된 아이템의 존재를 가리키도록 설정된다: 비트 0은 카트리지 1에 대응하고, 비트 1은 카트리지 2에 대응하며, 비트 2는 선택 CD 드라이브에 대응한다. 그 외 다른 비트는 0으로 설정된다.

또한, CPU는 정보를 I/O 코프로세서(36)에 기입함으로써 직렬 링크(22)를 경유해서 데이터를 I/O 장치들에 전송할 수 있다. 데이터 바이트들은 바이트 03H 만큼을 앞에 두고 I/O 포트 AS0에 기입된다. I/O 코프로세서는 이들 바이트를 I/O 장치들에 기입한다. 이러한 능력은, 예를 들어, 프린터(도시 안됨)에 데이터를 전송하는데 이용된다.

본 발명의 인스크라이브된 디폴트 템플리트 그래픽 이미지를 내장한 터치패드와 두 개의 조이스틱을 갖는 입력 장치에 인터페이스 하는 것 또한 자명하다. 제4도에 대한 부분에서 설명되는 것처럼, 시스템 BIOS로부터 CPU(30)상에서 실행되는 인터럽트 조정기(handler)는 I/O 코프로세서(36)를 경유해서 입력 장치들로부터 데이터를 수신한다. 인터럽트 조정기는 단지 I/O 코프로세서(36)로부터 임의의 전송들을 메모리(33)내로 배치한다. CPU(30)에 대해서 실행되는 응용 프로그램들은 입력들이 수신되었는지를 결정하기 위하여 소프트웨어 인터럽트를 경유해서 주기적으로 연산 시스템 BIOS를 폴(Poll)한다. 그후, 그들은 소프트웨어 인터럽트에 응답해서 연산시스템에 의해 응용 프로그램들에 통신된다.

응용 프로그램들은 현행 템플리트를 모니터한다. 디폴트 템플리트 그래픽 이미지가 검출되면 (오버레이 센서(124)가 모든 전송-형태 식별 패턴(108), 즉 000000를 검출할 것이다), 응용 프로그램은 디폴트 템플리트에 응해서 동작한다. 한편, 템플리트 오버레이(102)가 검출되면, 응용 프로그램은 특정 템플리트 오버레이(102)에 응해서 동작한다.

CPU(30)은 터치패드와 인터페이싱 하기 위해 특별한 지원을 하며 CPU상에서 실행되는 운영 체제를 지닌다. 이 운영 체제는 시스템 메모리(33) 내의 ROM이나 카트리지 ROM(40)으로부터 로딩된다. 운영 체제는 CPU(30)상에서 실행되는 응용 프로그램에 의해 호출될 수 있는 define_a_region, set_mapping_units, clear_all_regions, 및 interpret_a_point와 같은 명령을 가진다.

define_a_region 명령은 응용 프로그램이 터치패드의 한 영역을 정의하고 그 영역을 소정의 영역 식별자(region_id)와 연관시키도록 한다. 이 정의된 영역 내에 어떤 지점이 접속되더라도 그 region_id에 의해 식별될 것이다 한 실시예에서, 초기화 후에 터치패드 전제는 널(null) 영역 식별자(null_id)를 갖는 널 영역(null region)으로 정의되어진다. 즉, 어떤 null_id 위치가 접촉되더라도 운영 체제는 아무런 작업이나 기능을 수행하지 않는다. 즉, 어떠한 파라미터도 응용 프로그램에 전달되지 않는다. 또는, 예를들면 시스템으로 하여금 들을 수 있는 소리나 삑삑(beep) 소리를 내어 널 영역(null region) 이 접촉되었음을 사용자에게 알리는 에러처리 루틴(error-handling routine)을 트리거(trigger) 할 수 있다.
define_a_region 명령은 선택된 모양의 영역을 다른 region_id의 영역에 할당한다.

원, 사각형, 삼각형 등 거의 어떤 모양이나 어떤 모양의 조합도 define_a_region 명령에 의해 정의될 수

있다. 게다가, 다수의 모양들이 조합되어 복합 모양 영역(complex shaped region)을 형성할 수 있다. 예를 들어, 5개의 삼각형이 정5각형에 조합되어 별을 형성할 수 있다. 또한, define_a_region 명령을 사용하여 널 영역이 모양을 만드는데 포함될 수 있으며, 이에 따라 고리형(annuli)과 같은 중공형(hollow shapes)이 정의 될수 있다.

set_mapping_units 명령은 응용 프로그램이 define_a_region 명령에 대해 다른 유니트를 정의할 수 있게 한다. 이로 인하여, 운영 체제는 해상도가 다른 터치패드를 지원할 수 있다. 예를 들면, 특별한 명령이 없는 경우 운영 체제는 4각형 장치-해상도 유니트(rectangular device-resolution unit)를 정의하며, 이는 열과 행의 해상도(넓이와 간격)에 의해 결정된다. set_mapping_units 명령을 사용함으로써, 응용 프로그램은 운영 체제가 유니트를 예를 들어, 밀리미터나 100분의 1인치로 변화시키도록 할 수 있다. 또한, 응용 프로그램은 유니트를 다른 값으로 설정할 수 있다. 예를 들면, 시스템 디스플레이 장치가 640×480 화소의 해상도를 가진다면, 응용 프로그램은 set_mapping_units 명령을 사용하여, 터치패드 유니트를 화소들에 대응하는 640×480 유니트에 할당한다.

clear_all_regions 명령은 define_a_region 명령으로 미리 정의된 모든 영역을 소거하며, 실질적으로 전체 터치패드를 널 영역으로 정의한다. 응용 프로그램은 새로운 템플리트 오버레이가 터치패드 표면상에 놓여졌을 때, 이 명령을 호출하고, 그로 인하여 전의 템플리트 오버레이과 관련하여 정의된 영역을 소거한다. Clear_a_region 명령으로 영역을 소거(clear)한 후에, 응용 프로그램은 새로운 템플리트 오버레이 영역을 정의하는 define_a_region 명령을 호출하게 된다.

interpret_a_point 명령은 운영 체제가 접촉된 영역의 region_id를 결정하도록 한다. 이 명령은 운영 체제가 터치패드의 접촉을 모니터링(monitoring) 한다면 운영 체제에 의해 시작되고, 응용 프로그램이 터치패드의 접촉을 모니터링 한다면 응용 프로그램에 의해 시작된다. 한 실시예에서, 응용 프로그램은 운영 체제를 폴링하여 (polls) 운영자가 터치패트 영역을 접촉했는지 여부를 알아보도록 한다.

터치패드는 터치패드가 접촉되는 때마다 CPU에 장치 특정 좌표 정보(device-specific coordinate information)을 보낸다. interpret_a_point 명령의 일부는 CPU에 의한 접촉된 영역의 결정이다. 이 결정 후에, 운영 체제는 접촉된 영역의 region_id나 어떤 영역도 접촉되지 않았음을 나타내는 코드에 상응하여 코드를 응용 프로그램에 되돌려보낸다.

운영 체제에서 터치패드를 지원하는 컴퓨터 시스템의 사용은 매우 간단하다. 우선, 응용 프로그램은 define_a_region 명령을 사용하여 터치패드 영역을 정의하고, region_id를 영역과 관련시킨다. 정의될 각각의 영역은 define_a_region 명령을 사용하여 정의 되어야 한다. 아무것도 정의 되지 않은 영역은 null_id를 가진 null 영역으로 남아있을 것이다.

후속하는 터치패드 표면의 접촉은 어느 쪽이 터치패드를 모니터링하는 지에 따라 운영 체제 또는 응용 프로그램에 의해 검출되어진다. 운영 체제가 터치패드를 모니터링한다면, 접촉을 검출하고, interpret_a_point 명령을 개시하고, 이는 접촉된 영역의 region_id를 결정한다. 만약, region_id 가 null_id라면, 운영 체제는 삑삑소리를 내는 등의 적절한 동작을 수행하게 된다. 이 경우에, 어떠한 파라미터도 응용 프로그램으로 전달되지 못한다. 만약 region_id가 null_id가 아니면, 운영 체제는 접촉된 영역과 관련된 region_id를 응용 프로그램에 전달하고, 응용 프로그램은 식별된 region_id에 관련된 기능을 수행한다. 만약 응용 프로그램이 터치패드를 모니터링하고 터치패드의 접촉을 검출한다면, 운영 체제가 명령을 개시하지 않고 응용 프로그램이 운용 체제로 하여금 interpret_a_point 명령을 개시하도록 한다는 점을 제외하고는 동일한 과정이다.

만약 새로운 템플리트 오버레이가 터치패드 표면에 놓여지면, 응용 프로그램은 clear_all_region 명령을 호출하고, 그것은 운영 체제로 하여금 다시 한번 모든 정의된 영역을 소거하고, 전체적인 터치패드 표면을 널 영역으로 간주하도록 한다. 다시 한번, 응용 프로그램은 운영 체제로 하여금 응용 프로그램에 의해 지시된 대로 템플리트의 적절한 영역을 정의하도록 하고 region_id를 각 영역들에 할당하도록 하기 위해 define_a_region 명령을 부른다.

정의된 모양의 영역을 정의하고 매핑하는 특정한 예는 제4도에 도시되어 있고, 여기서는 매핑된 후의 터치패드가 도시되어 있다. 감응 영역(sensitive areas)의 영역은 (1) 전체적으로 참조 번호(196)로 표시된 화살표 모양과, (2) 전체적으로 참조 번호 (198)로 표시되고, 고리 모양의 널 영역에 의해 분리된 원과 고리 모양으로 나타난다. 개별적인 감응 영역은 아래에서 식별되어지듯 분리된 작은 심볼들로 도시된다. 초기화 또는 clear_all_regions 명령이 수행된 후에, 터치패드의 모든 감응영역은 null_id에 할당하고 이는 · 심볼에 의해 제4도에 도시되어 있다. 이것은 ·가 접촉된 지역에 가장 밀접해 있다면, 그 영역이 null_id로 할당되어지고 운영체제는 상술한 대로 반응한다는 것을 의미한다.

고리형 널 영역에 의해 분리된 원과 고리를 만들기 위해, 응용 프로그램은 define_a_region 명령을 세 번 호출하여야 한다. 우선, 응용 프로그램은 (circle, 14, F, 4, 1)를 파라미터로 하여 define_a_region 명령을 호출한다. circle이라는 용어는 그려질 모양을 나타낸다. 14,F는 원의 중심 좌표를 나타낸다. 4는 현재 유니트, 즉 사각형 장치-해상도 유니트에서 원의 반지름을 나타낸다. 그리고, 1은 그 특정 원에 할당된 region_id를 나타낸다. 이러한 명령은 제4도에서 원(200)에 의해 둘러싸인 52개의 감응 영역(sensitive area)를 포함하는 가상의 원(200)을 만들 것이다. 이러한 52개의 감응 영역은 모두 region_id 1로 할당되어 있고, 제4도에서 ⦿. · 그리고 ○ 의 심볼로서 도시된다. ⦿ 심볼로 표시된 28개의 감응 영역은 region_id 1에 할당된 채로 남아 있다. 나머지는 이어지는 define_a_region 명령에 의해 다른 region_id로 재 할당된다.

다음, 응용 프로그램은 (circle, 14, F, 3, 0)를 파라미터로 하여 define_a_region 명령을 호출하는데, 여기서는 0은 region_id로서 null_id를 나타낸다. 이것은 원(202)에 의해 둘러싸인 24개의 감응 영역을 포함하는 가상의 원(202)를 만든다. 이러한 24개의 감응 영역은 모두 region_id 0으로 할당되고 제4도에서 ·과 ○ 심볼에 의해 도시된다. 이 24개의 감응 영역은 이전의 define_a_region 명령으로 region_id 1에 할당되었다는 점을 상기해야 한다. 그래서, 이것은 region_id 0으로 재할당 된 것으로도 간주될 수 있다. · 심볼로 도시된 12개의 감응 영역 region_id 0으로 할당된 채로 있다. ○ 심볼로 도시된 12개의 감응 영역은 제3의 define_a_region 명령에 의해 다른 region_id로 재 할당 될 것이다. ·심볼로 도시된

12개의 감응 영역은 상술한 대로 널 영역이다.

마지막으로, 고리형 널 영역에 의해 분리된 원과 고리를 완성하기 위해, 최종적으로 (circle, 14, F, 3, 2)를 파라미터로 하여 define_a_region 명령을 호출한다. 이 명령은 ○ 심볼에 의해 도시된 12개의 점들을 포함하는 가상의 원 (204)를 만든다. 이 12개의 감응 영역은 region_id 2로 할당된다. 이러한 특정 12개의 감응 영역은 제1 define_a_region 명령에 의해 region_id 1로 할당되었고, 그리고 나서 이전의 define_a_region 명령에 의해 null_id로 재할당되었다는 점을 상기해야 한다.

결과적인 복합 모양(198)은 (region_id 0에 할당된) 고리형 널 영역(210)에 의해 분리된, region_id로 2로 할당된 채워진 원(206)과 region_id 1로 할당된 고리(208)이다. 채워진 원(206)에서 감응 영역은 ○ 심볼로 도시된다. 고리(208)에서 감응 영역은 ⦿심볼로 도시된다. 마지막으로, 고리형 널 영역(210)에서 감응 영역은 · 심볼로 도시된다.

한편, define_a_region 명령은 곧바로 고리를 지원하도록 형성될 수 있고, 그로 인하여 복합 모양(198)이 원과 고리의 두 명령으로 만들어 질 수 있다.

화살표(196)을 만드는 경우에는 define_a_region 명령을 두 번 호출하여야 한다. 제1 호출의 경우에는 (rectangle, 6, C, 9, I, 3)의 파라미터를 가지는데, rectangle은 모양을, 6, C는 직사각형의 좌측 아래쪽 꼭지점을 9, I는 직사각형의 우측 위쪽 꼭지점을 3은 region_id를 나타낸다. 이러한 명령은■ 심볼로 도시되는 감응 영역에 둘러싸인 가상의 직사각형을 만든다. 제2 define_a_region 명령은 (triangle, 1, F, 6, A, 6, K, 3)의 파라미터를 가지는데, triangle은 모양을 1, F, 6, A와 6, K는 3개의 꼭지점을 나타나며, 3은 region_id를 타나낸다. 이러한 명령은 ▲ 심볼로 도시된 감응 영역을 둘러싸는 가상의 삼각형(214)를 만든다. 직사각형(212)와 삼각형 모두가 region_id 3에 할당된다는 사실을 주목해야 한다. 양자는 같은 region_id를 갖는다. 그래서, 두 모양은 같은 기능으로 매핑되고 화살표(196)을 구성한다. 그리하여, 한 지역 또는 양 지역 어느 쪽에든 접촉이 있으면 응용 프로그램에서 같은 반응을 하게 된다. 남아 있는 터치패드 표면의 감응 영역은 일반적으로 참조 번호(215)로 도시되는데, 여전히 region_id로서 null_id가 할당되어 있다. 이러한 감응 영역은 · 심볼로 도시된다.

매우 다양한 복합 모양일지라도 이러한 방식으로 형성될 수 있다는 것은 이제 명백하다. 각각의 모양은 유일한 region_id를 갖거나, 혹은 하나 이상의 영역이 region_id를 공유할 수 있다.

이용하는 동안에, 터치패드에서 제4도에 도시된 매핑된 영역의 터치패드를 누르는 것은 운영 체제로 하여금 영역의 region_id를 결정하도록 한다. 예를 들어 (참조번호, 216으로 도시된) 위치 13, E를 누르면 접촉된 위치의 좌표가 링크(22)를 거쳐 CPU에 전달되고, 운영 체제는 region_id 2가 눌러졌음을 결정하여, region_id 2를 응용 프로그램에 전달한다. 그리고 나서, 응용 프로그램은 region_id2에 관계된 기능을 실행한다.

반면에, null-id로 할당된 영역을 누르면 null-id는 운영 체제로 하여금 region_id를 응용 프로그램에 전달하도록 하지는 않는다. 오히려, 상술한대로 운영체제는 아무일도 하지 않거나 삑 소리를 내거나 혹은 다른 적절한 행동을 한다. 응용 프로그램이 널 영역을 접촉했는지 검출하기 위해서는, 응용 프로그램은 전체적인 터치패드 표면을 어떤 region_id로 할당하기 위해 define_a_region 명령을 우선 호출하고, 중공형 영역(hollow region)을 만들 때에 중공 영역을 그 region_id로 할당한다.

상기의 매핑에서 가상적(virtual)이란 용어는 원(200), 원(202), 원(204), 직사각형(212) 그리고 삼각형(214)와 같은 모양을 표현할 때 사용된다. 이 용어는 감응 영역이 무한대의 해상도를 갖지 못하기에 사용되는 것이다. 사실, 다양한 심볼에 의해 정의된 감응 영역이란 용어가 사용될 때, 그 용어는 그러한 심볼에 가장 근접한 표면의 지역을 의미한다. 따라서, 어떠한 정의된 모양도 감응 영역의 영역이 매핑하는 최종적인 모양에 근사하는 것일 뿐 이다. 터치패드의 해상도가 높아질수록, 감응 영역의 매필은 그것을 정의하는 가상의 모양에 더욱 가까이 근사하게 되는 것이다.

본 발명이 실시예의 설명에 의해 예시되고 실시예들이 상당히 상세하게 설명되었지만, 첨부된 특허 청구의 범위를 이러한 상세한 설명에 한정하거나 어떤 방식으로든 제한하는 것은 출원인의 의도가 아니다. 본 발명의 부가적인 장점이나 변형은 본 기술분야에서 통상의 지식을 가진 자라면 용일할 것이다.

그러므로, 넓은 면에서의 본 발명은 도시되고, 기술된 특정 세부 사항, 표현된 장치 및 방법, 그리고 예시적인 실시예에 한정되지 않는다. 따라서, 출원인의 총괄적 발명 개념의 사상이나 범위를 벗어남이 없이 그러한 세부적인 사항으로부터 벗어날 수 있을 것이다.

(57) 청구의 범위

청구항 1

컴퓨터 시스템에 있어서, 운영 체제 및 응용 프로그램을 실행하기 위한 중앙 처리 장치(CPU). 상기 CPU와 회선으로 통신하며(in circuit communication) 운영 체제 및 응용 프로그램을 저장하기 위한 메모리, 적어도 하나의 외부 입력장치 - 상기 입력 장치는 압력에 감응하는 경계 평면(pressure sensitive bounded plane)을 가지며 상기 평면의 경계내의 임의의 위치에 가해지는 국부적인 압력을 검출하기 위함 것임 - 로부터의 좌표 타입의 데이터를 상기 CPU에 인터페이스시키기 위하여 상기 CPU와 회선으로 통신하는 주변 인터페이스 회로(peripheral interface circuit), 및 상기 CPU 및 상기 주변 인터페이스 회로와 연관되어 있고, 상기 CPU에서 실행되는 적어도 하나의 응용 프로그램으로부터의 입력에 응답하여 상기 입력 장치의 경계 평면의 영역을 동적으로 정의하고 상기 정의된 영역을 영역 식별자(region identifier)와 연관시키으로써 상기 CPU에 의한 운영 체제 및 응용 프로그램의 실행에 응답하는 접촉 영역 정의 로직(touch region definition logic) - 상기 접촉 영역 정의 로직은 상기 영역의 생성을 위해 가변 영역 크기에 관련된 영역 정의 입력을 받아들임 - 을 포함하는 컴퓨터 시스템.

청구항 2

제1항에 있어서, 상기 CPU와 연관되어 있으며, 상기 입력 장치의 경계 평면의 접촉에 응답하여 상기 입력 장치의 경계 평면의 접촉된 영역에 대한 영역 식별자를 결정함으로써 상기 CPU에 의한 운영 체제 및 응용 프로그램의 실행에 응답하는 영역 식별 로직(region identification logic)을 더 포함하는 컴퓨터 시스템.

청구항 3

제1항에 있어서, 상기 접촉 영역 정의 로직은 상기 입력 장치의 경계 평면의 적어도 하나의 널 영역(null region)을 동적으로 정의하되, 상기 적어도 하나의 널 영역의 접촉이 상기 적어도 하나의 응용 프로그램에 전달되지 않도록 정의함으로써 상기 CPU에 의한 운영 체제 및 응용 프로그램의 실행에 응답하는 컴퓨터 시스템.

청구항 4

제1항에 있어서. 상기 CPU는 운영 체제 및 응용 프로그램을 실행하는 동안 상기 입력 장치의 경계 평면을 매핑된 영역 유니트(mapped units of area)로 분할하고, 상기 접촉 영역 정의 로직은 상기 매핑된 유니트에 해당하는 단위로(in units) 영역 정의 입력 (region definition inputs)을 받아들이며, 상기 CPU와 연관되어 있고, 상기 접촉 영역 정의 로직으로의 상기 입력의 매핑된 유니트를 상기 적어도 하나의 응용 프로그램으로부터의 입력에 의해 지시된 바에 따라 변경시킴으로써 상기 CPU에 의한 운영 체제 및 응용 프로그램의 실행에 응답하는 터치패드 매핑 유니트 로직(touchpad mapping logic)을 더 포함하는 컴퓨터 시스템.

청구항 5

제1항에 있어서, 상기 접촉 영역 정의 로직은 상기 CPU와 회선으로 통신하는 회로를 포함하는 컴퓨터 시스템.

청구항 6

제1항에 있어서, 상기 접촉 영역 정의 로직은 상기 CPU에서 실행하기 위한 실행가능 코드(executable code)를 포함하는 컴퓨터 시스템.

청구항 7

제1항에 있어서, 상기 접촉 영역 정의 로직은 상기 영역의 생성을 위해 가변 영역 형태에 관련된 영역 정의 입력을 받아들여서, 다양한 기하학적 모양을 갖는 영역을 정의 할 수 있도록 하는 컴퓨터 시스템.

청구항 8

제1항에 있어서, 상기 접촉 영역 정의 로직은 상기 영역의 생성을 위해 가변 영역 형태에 관련된 영역 정의 입력을 받아들여서, 삼각형, 사각형 그리고 원형 영역을 정의할 수 있도록 하는 컴퓨터 시스템.

청구항 9

컴퓨터 시스템에 있어서, (a) 운영 체제와 응용 프로그램을 실행하기 위한 중앙 처리 장치(CPU), (b) 상기 CPU와 회선을 통신하며, 운영 체제와 응용 프로그램을 저장하기 위한 메모리 회로, (c) 상기 CPU 및 상기 메모리와 회선으로 통신하여 비디오 디스플레이 장치에 디스플레이될 시각적 영상에 대응하는 전기 신호를 발생시키기 위한 비디오 회로, (d) 상기 CPU와 회선으로 통신하며, 상기 CPU에 외부 장치로부터의 신호를 인터페이스시키기 위한 주변 인터페이스 회로, (e) 국부적 압력을 가하기 위한 부재(member for applying localized pressure)에 의해 접촉할 수 있도록 노출되어 있는 패드 표면, (f) 템플리트 오버레이(template overlay)를 상기 패드 표면에 근접하도록 착탈가능하게 고정시키기 위한 리테이너 (retainer for removably securing), (g) 상기 패드 표면에 근접하게 배치되어, 적어도 하나의 전기 신호를 발생시키고, 상기 전기 신호들의 집단(cumulation of said electrical signals)이 상기 패드 표면을 접촉하거나 상기 패드 표면에 근접해 있는 템플리트 오버레이를 접촉하는 상기 부재의 위치의 좌표들에 대응하도록 구성된 좌표 센서, (h) 상기 패드 표면에 근접한 강성(rigid) 베이스, (i) 상기 좌표 센서와 회선으로 통신하며, 상기 패드 표면을 접촉하거나 상기 패드 표면에 근접해 있는 템플리트 오버레이를 접촉하는 상기 부재의 좌표들을 결정하기 위한 좌표 결정회로, (j) 상기 좌표 결정 회로 및 상기 주변 인터페이스 회로와 회선으로 통신하며, 상기 결정된 좌표를 상기 좌표 결정 회로 및 상기 주변 인터페이스 회로로 전달(communicate)하기 위한 인터페이스 회로, (k) 상기 CPU 및 상기 주변 인터페이스 회로와 연관되어 있고, 상기 CPU에서 실행되는 적어도 하나의 응용 프로그램으로부터의 입력에 응답하여 상기 패드표면의 영역을 동적으로 정의하고 상기 정의된 영역을 영역 식별자(region identifier)와 연관시킴으로써 상기 CPU에 의한 운영 체제 및 응용 프로그램의 실행에 응답하는 접촉 영역 정의 로직(touch region definition logic) - 상기 접촉 영역 정의 로직은 상기 영역의 생성을 위해 가변 영역 크기에 관련된 영역 정의 입력을 받아들임 -을 포함하는 컴퓨터 시스템.

청구항 10

제9항에 있어서, 상기 CPU와 연관되어 있고, 상기 패드 표면의 접촉에 응답하여, 상기 패드 표면의 접촉된 영역에 대한 상기 영역 식별자를 결정함으로써 상기 CPU에 의한 운영 체제와 응용 프로그램의 실행에 응답하는 영역 식별로직(region identification logic)을 더 포함하는 컴퓨터 시스템.

청구항 11

제9항에 있어서, 상기 접촉 영역 정의 로직은 상기 패드 표면의 적어도 하나의 널 영역을 동적으로 정의하되 상기 적어도 하나의 널 영역의 접촉이 상기 적어도 하나의 응용 프로그램에 전달되지 않도록 정의함으로써, 상기 CPU에 의한 운영 체제 및 응용 프로그램의 실행에 응답하는 컴퓨터 시스템.

청구항 12

제9항에 있어서, 상기 CPU는 운영 체제와 응용 프로그램을 실행하는 동안 상기 입력 장치의 경계 평면을 매핑된 영역 유니트로 분할하고, 상기 접촉 영역 정의 로직은 상기 매핑된 유니트에 해당하는 단위로 영역 정의 입력을 받아들이며, 상기 CPU와 연관되어 있고, 상기 접촉 영역 정의 로직으로서의 상기 입력의 매핑된 유니트를 상기 적어도 하나의 응용 프로그램으로부터의 입력에 의해 지시된 바에 따라 변경시킴으로써 상기 CPU에 의한 운영 체제와 응용 프로그램의 실행에 응답하는 터치패드 매핑 유니트 로직을 더 포함하는 컴퓨터 시스템.

청구항 13

제9항에 있어서, 상기 접촉 영역 정의 로직이 상기 CPU와 회선으로 통신하는 회로를 포함하는 컴퓨터 시스템.

청구항 14

제9항에 있어서, 상기 접촉 영역 정의 로직이 상기 CPU 상에서 실행되기 위한 실행 가능 코드를 포함하는 컴퓨터 시스템.

청구항 15

운영 체제 및 응용 프로그램을 실행하기 위한 중앙 처리 장치(CPU)와, 상기 CPU와 회선으로 통신하며, 접촉 감응 표면(touch-sensitive surface)을 갖는 불투명한(opaque) 좌표 타입의 입력 장치(coordinate type input device)를 CPU에 전기적으로 인터페이스 시키기 위한 주변 인터페이스 회로를 구비하는 컴퓨터 시스템에서, 상기 운영체제의 프로그램은 상기 CPU에 의해 엑세스될 수 있도록 저장되고, 상기 CPU에서 실행될 때 상기 CPU와 함께 동작하며, 상기 운영 체제 프로그램은 (a) 상기 CPU에서 실행되는 적어도 하나의 응용 프로그램으로부터의 입력에 응답하여, 좌표 타입의 입력 장치의 적어도 하나의 영역을 동적으로 정의하고 상기 영역을 영역 식별자와 연관시키도록 구성된 영역 정의 루틴(region defining routine) - 상기 영역 정의 루틴은 상기 영역의 생성을 위해 가변 영역 크기에 관련된 영역정의 입력을 받아들임-,(b) 상기 주변 인터페이스 회로로부터 좌표 타입의 데이터를 획득하도록 구성된 터치패드 인터페이스 루틴, (c) 상기 접촉 감응 표현의 접촉에 응답하여, 상기 좌표 타입의 입력 장치의 영역에 대한 상기 영역 식별자를 결정하도록 구성된 영역 식별 루틴(region identification routine), (d) 상기 적어도 하나의 응용 프로그램에 상기 결정된 영역 식별자를 전달하도록 구성된 영역 식별 전달 루틴(region identity communication routine)을 포함하는 컴퓨터 시스템.

청구항 16

제15항에 있어서, (a) 상기 영역 정의 루틴은 상기 적어도 하나의 응용 프로그램에 응답하여 상기 접촉 감응 표현의 적어도 하나의 널 영역을 동적으로 정의할 수 있는 것을 부가적 특징으로 하며, 상기 적어도 하나의 널 영역은 상기 널 영역을 접촉한 것이 상기 적어도 하나의 응용 프로그램에 전달되지 않는 것을 특징으로 하며, (b) 상기 영역 식별 전달 루틴은 상기 적어도 하나의 널 영역의 접촉을 상기 적어도 하나의 응용 프로그램에 전달하지 않는 것을 특징으로 하는 컴퓨터 시스템.

청구항 17

제15항에 있어서, 상기 운영 체제 프로그램 및 상기 CPU는 상기 입력 장치의 상기 접촉 감응 표면을 매핑된 영역 유니트로 분할하기 위해 함께 동작하며, 상기 영역 정의 루틴은 상기 매핑된 유니트에 해당하는 단위로 입력을 받아 들이며, 상기 영역 정의 루틴으로서 상기 입력의 매핑된 유니트를 상기 적어도 하나의 응용 프로그램으로부터의 입력에 의해 지시된 바에 따라 변경시키도록 구성된 터치패드 매핑 유니트 루틴을 더 포함하는 컴퓨터 시스템.

청구항 18

제9항에 있어서, 상기 CPU와 연관되어 있고, 상기 패드 표면의 접촉에 응답하여 상기 패드 표면의 접촉된 영역에 대한 상기 영역 식별자를 결정함으로써 상기 CPU에 의한 운영 체제와 응용 프로그램의 실행에 응답하는 영역 식별 로직(region identification logic)을 더 포함하며, 상기 접촉 영역 정의 로직은 상기 패드 표면의 적어도 하나의 널 영역을 동적으로 정의하되 상기 적어도 하나의 널 영역의 접촉이 상기 적어도 하나의 응용 프로그램에 전달되지 않도록 정의함으로써 상기 CPU에 의한 운영 체제 및 응용 프로그램의 실행에 응답하는 컴퓨터 시스템.

청구항 19

제18항에 있어서, 상기 CPU는 운영 체제 및 응용 프로그램을 실행하는 동안 상기 입력 장치의 경계 평면을 매핑된 영역 유니트(mapped units of area)로 분할하고, 상기 접촉 영역 정의 로직은 상기 매핑된 유니트에 해당하는 단위로 영역 정의 입력(region definition inputs)을 받아들이며, 상기 CPU와 연관되어 있고, 상기 접촉 영역 정의 로직으로의 상기 입력의 매핑된 유니트를 상기 적어도 하나의 응용 프로그램으로부터의 입력에 의해 지시된 바에 따라 변경시켜 상기 CPU에 의한 운용 체제 및 응용 프로그램의 실행에 응답하는 터치패드 매핑 유니트 로직을 더 포함하는 컴퓨터 시스템.

청구항 20

제1항에 있어서, 상기 CPU와 연관되어 있으며, 상기 입력 장치의 경계 평면의 접촉에 응답하여 상기 입력 장치의 경계 평면의 접촉된 영역에 대한 영역 식별자를 결정함으로써 상기 CPU에 의한 운영 체제 및 응용 프로그램의 실행에 응답하는 영역 식별 로직 (region identification logic)을 더 포함하며, 상기 접촉 영역 로직은 상기 입력 장치의 경계 평면의 적어도 하나의 널 영역(null region)을 동적으로 정의하되 상기 적어도 하나의 널 영역의 접촉이 상기 적어도 하나의 응용 프로그램에 전달되지 않도록 하는 상기 CPU

에 의한 운영체제 및 응용 프로그램의 실행에 응답하는 컴퓨터 시스템.

청구항 21

제20항에 있어서, 상기 CPU는 운영 체제 및 응용 프로그램을 실행하는 동안 상기 입력 장치의 경계 평면을 매핑된 영역 유니트(mapped units of area)로 분할하고, 상기 접촉 영역 정의 로직은 상기 매핑된 유니트에 해당하는 단위로 영역 정의 입력(region definition inputs)를 받아들이며, 상기 CPU와 연관되어 있고, 상기 접촉 영역 정의 로직으로서의 상기 입력의 매핑된 유니트를 상기 적어도 하나의 응용 프로그래으로부터의 입력에 의해 지시된 바에 따라 변경시킴으로써 상기 CPU에 의한 운영 체제 및 응용 프로그램의 실행에 응답하는 터치패드 매핑 유니트 로직을 더 포함하는 컴퓨터 시스템.

청구항 22

제1항에 있어서, 상기 접촉 영역 정의 로직은 상기 CPU에서 실행되는 적어도 하나의 응용 프로그램으로부터의 입력에 응답하여, 상기 입력 장치의 경계평면의 복수의 제1 영역을 동적으로 정의하고, 상기 복수의 제1 영역을 제1 영역 식별자와 연관시킴으로써 상기 CPU에 의한 운영 체제 및 응용 프로그램의 실행에 응답하는 컴퓨터 시스템.

청구항 23

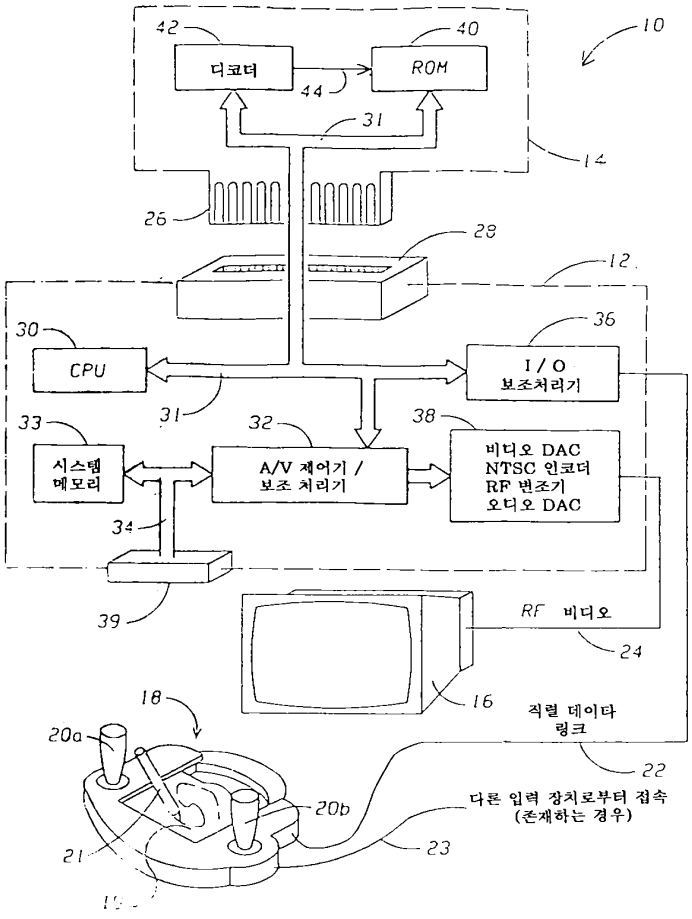
제22항에 있어서, 상기 접촉 영역 정의 로직은 상기 CPU에서 실행되는 적어도 하나의 응용 프로그램으로부터의 입력에 응답하여, 상기 입력 장치의 경계 평면의 복수의 제2 영역을 동적으로 정의하고, 상기 복수의 제2 영역을 제2 영역 식별자와 연관시키므로써 상기 CPU에 의한 운영 체제 및 응용 프로그램의 실행에 응답하는 컴퓨터 시스템.

청구항 24

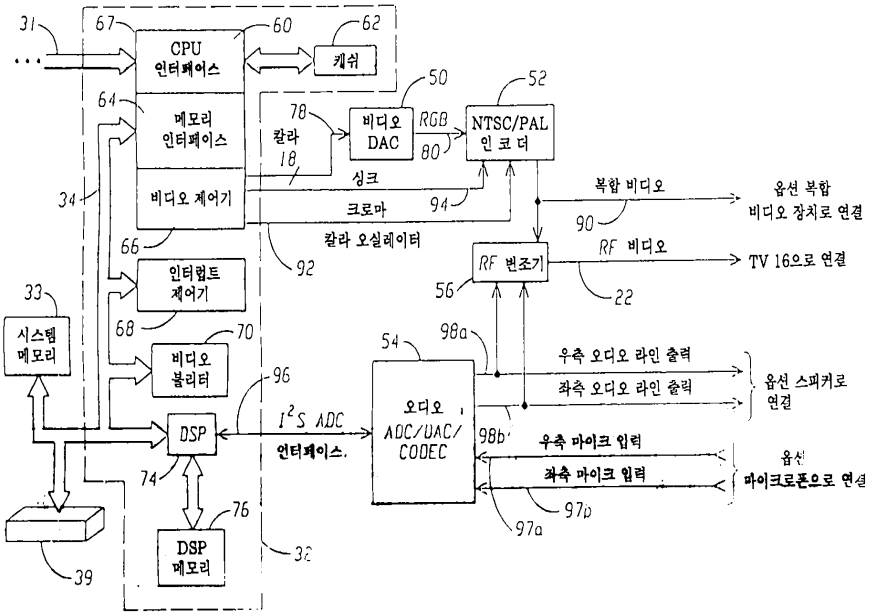
제3항에 있어서, 상기 접촉 영역 정의 로직은 전적으로 상기 정의된 영역의 경계내에서 상기 적어도 하나의 널 영역을 정의하는 컴퓨터 시스템.

도면

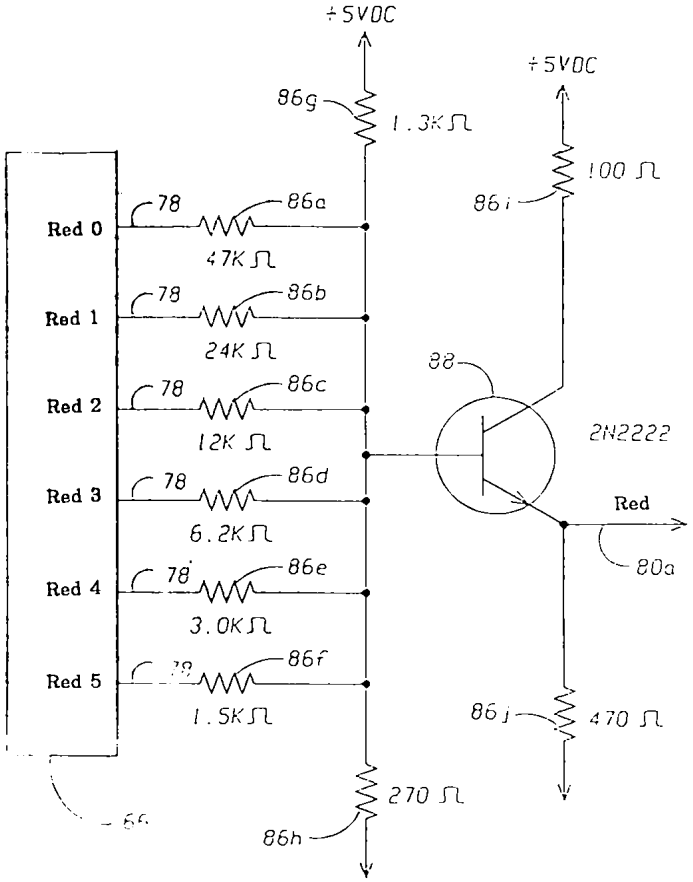
도면1a



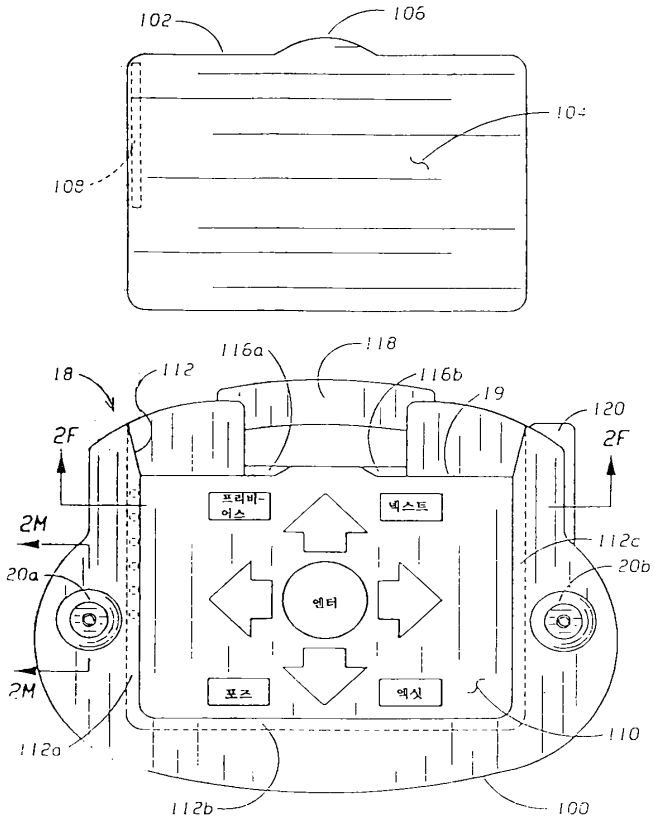
도면1b


CPU 인터페이스
캐쉬
메모리 인터페이스
비디오 제어기
인터럽트 제어기
비디오 불리터
DSP
DSP 메모리
시스템 메모리
칼라
비디오 DAC
RGB
NTSC/PAL 인 코 더
싱크
크로마
칼라 오실레이터
복합 비디오
옵션 복합 비디오 장치로 연결
RF 변조기
RF 비디오
TV 16으로 연결
I²S ADC 인터페이스
오디오 ADC/DAC/ CODEC
우측 오디오 라인 출력
좌측 오디오 라인 출력
옵션 스피커로 연결
우측 마이크 입력
좌측 마이크 입력
옵션 마이크로폰으로 연결

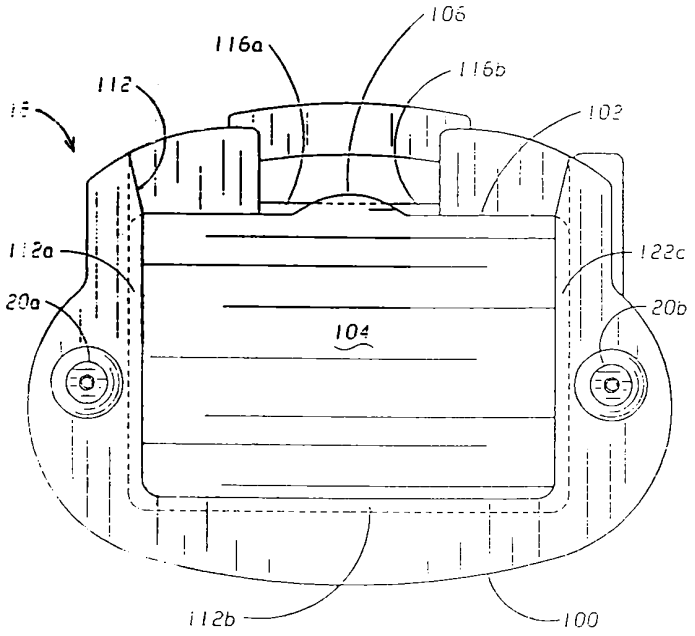
도면1c


+5VDC
+5VDC
1.3KΩ
100Ω
Red 0
Red 1
Red 2
Red 3
Red 4
Red 5
47KΩ
24KΩ
12KΩ
6.2KΩ
3.0KΩ
1.5KΩ
2N2222
Red
470Ω
270Ω

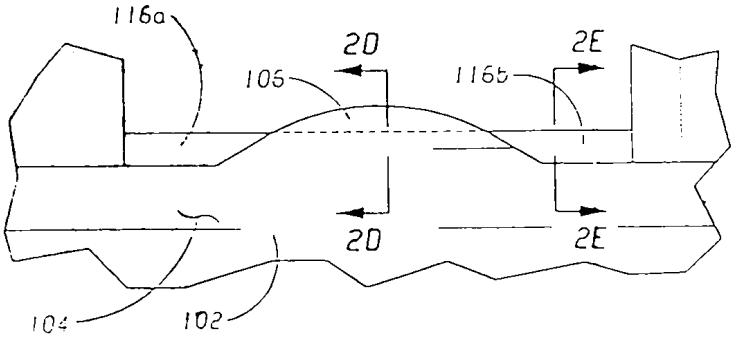
도면2a


102
106
104
108
18
112
116a
118
116b
19
120
2F
2F
2M
2M
20a
20b
112c
112a
112b
110
100
프리비-
어스
넥스트
엔터
포즈
엑싯

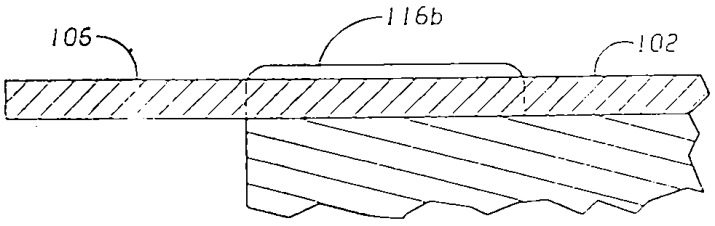
도면2b



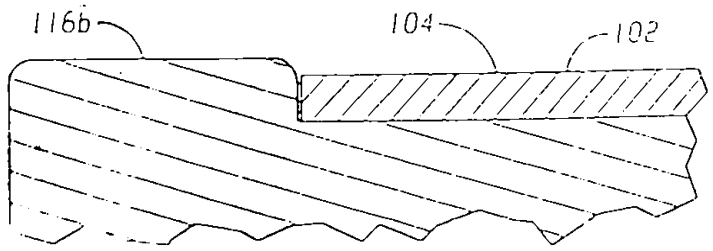
도면2c



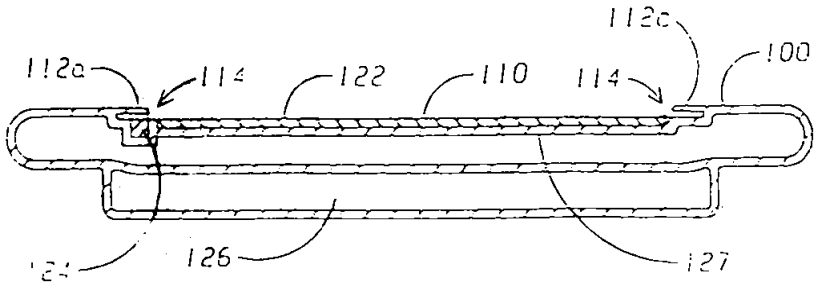
도면2d



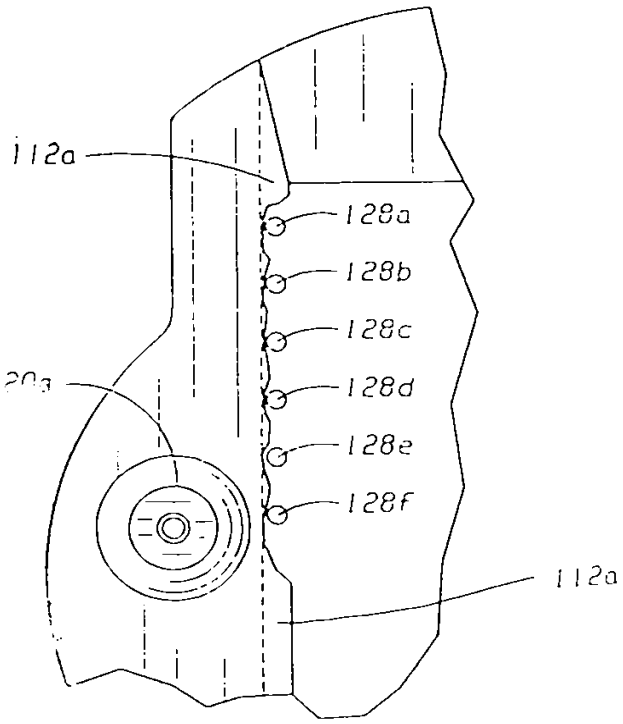
도면2e



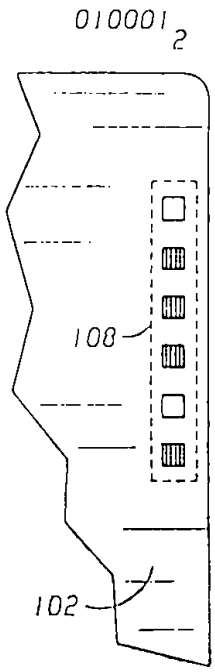
도면2f



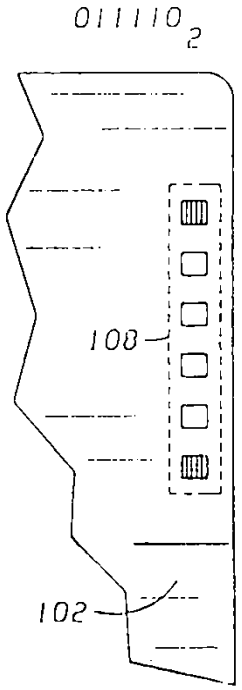
도면2g



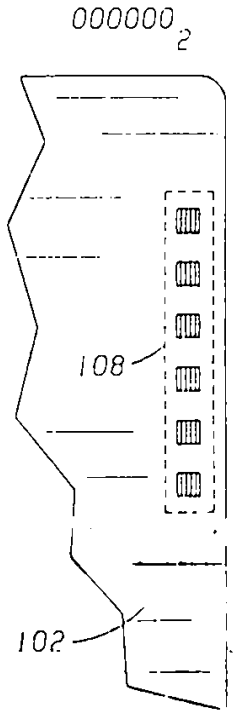
도면2h



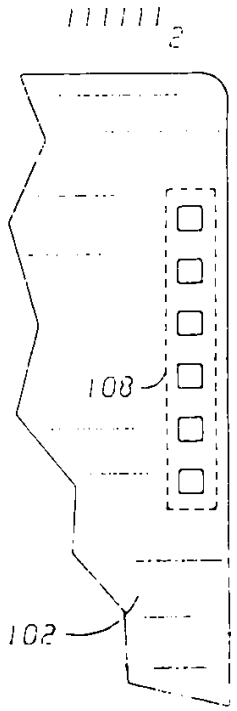
도면2i



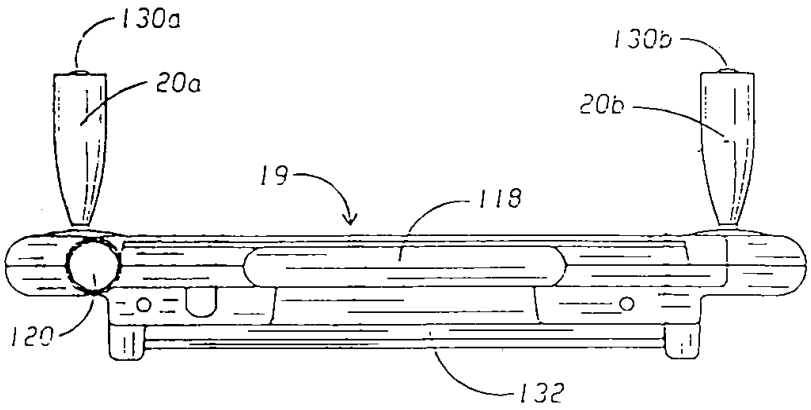
도면2j



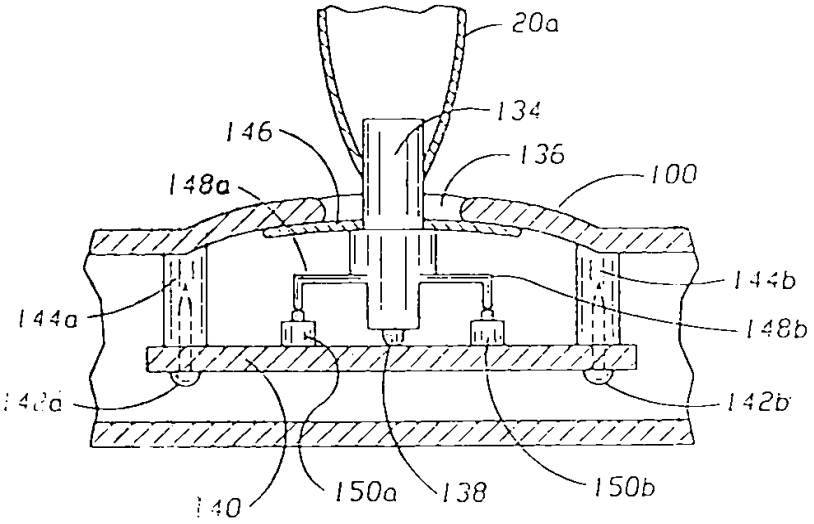
도면2k



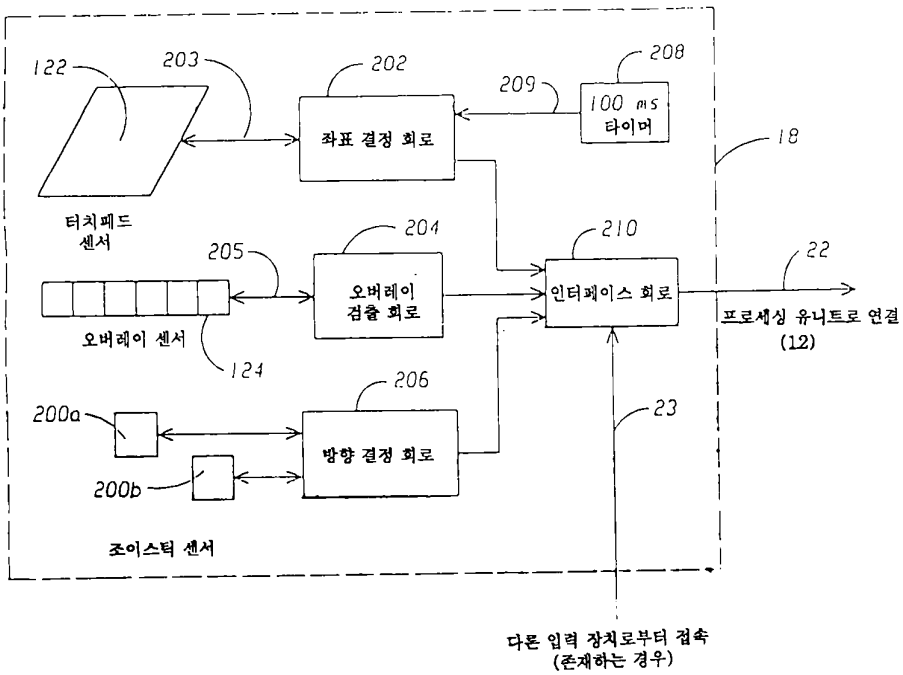
도면21



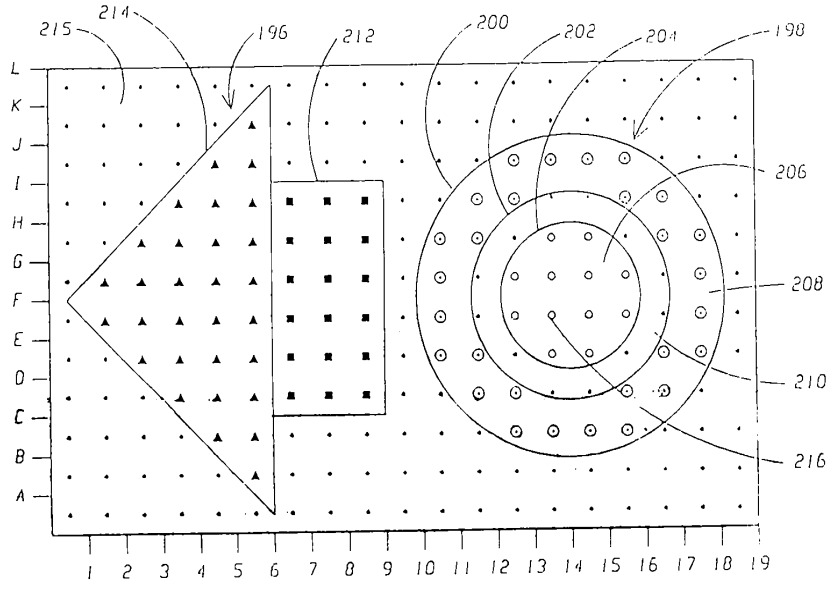
도면2m


20a
134
146
136
148a
100
144b
144a
148b
142a
142b
140
150a
138
150b

도면3


122
203
202
208
209
100 ms
타이머
좌표 결정 회로
18
터치패드
센서
205
204
210
22
오버레이
검출 회로
인터페이스 회로
프로세싱 유니트로 연결
(12)
오버레이 센서
124
206
23
200a
방향 결정 회로
200b
조이스틱 센서
다른 입력 장치로부터 접속
(존재하는 경우)

도면4


214
215
196
212
200
202
204
198
206
208
210
216
L
K
J
I
H
G
F
E
D
C
B
A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19