



(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。 Int. Cl.

H04B 1/40 (2006.01)

G09G 3/20 (2006.01)

G09G 5/393 (2006.01)

G09G 5/39 (2006.01)

(11) 공개번호 10-2007-0022236

(43) 공개일자 2007년02월26일

(21) 출원번호 10-2006-7019717

(22) 출원일자 2006년09월23일

심사청구일자 2006년09월23일

번역문 제출일자 2006년09월23일

(86) 국제출원번호 PCT/US2005/006327

(87) 국제공개번호 WO 2005/083672

국제출원일자 2005년02월24일

국제공개일자 2005년09월09일

(30) 우선권주장 10/954,408 2004년09월29일 미국(US)

60/547,711 2004년02월24일 미국(US)

60/569,288 2004년05월06일 미국(US)

(71) 출원인 쉘컴 인코포레이티드

미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775

(72) 발명자 루드윈 엘버트 스콧

미국 92121 캘리포니아주 샌디에고 오크뷰 웨이 11975

킹 스콧 하워드

미국 92064 캘리포니아주 포웨이 올드 포메라도 로드 12060

(74) 대리인 특허법인코리아나

전체 청구항 수 : 총 33 항

(54) 무선 디바이스용 디스플레이 프로세서

(57) 요약

디스플레이 프로세서는 인터페이스 유닛, 명령 프로세서, 동기화 유닛, 하나 이상의 프로세싱 유닛, 및 디바이스 버퍼를 포함한다. 인터페이스 유닛은 입력 이미지 데이터를 수신하고 (예를 들어, 메인 메모리로부터) 프레임 버퍼에 대해 출력 이미지 데이터를 제공한다. 명령 프로세서는 명령들을 수신하고 (예를 들어, 스크립트 또는 리스트로) 프로세싱 유닛(들)의 동작을 지시한다. 동기화 유닛은 LCD 스크린상의 인위적 영상을 유발하는 것을 방지하기 위해 프레임 버퍼에 대한 판독 포인터의 위치를 결정하고 프레임 버퍼에 출력 이미지 데이터의 기록을 제어한다. 프로세싱 유닛(들)은 영역 뒤집기, 영역 회전, 2 개의 비디오 포맷 사이의 색 변환 (예를 들어, YCrCb 로부터 RGB 로), 업/다운 이미지 사이즈 리스케일링, 알파-블렌딩, 투명성, 텍스트 오버레이 등과 같은 다양한 포스트-프로세싱 기능을 수행할 수도 있다.

대표도

도 1

특허청구의 범위

청구항 1.

입력 이미지 데이터를 수신하고 전자 스크린상의 프리젠테이션을 위한 출력 이미지 데이터를 제공하도록 동작 가능한 인터페이스 유닛; 및

상기 입력 이미지 데이터에 대해 하나 이상의 포스트-프로세싱 기능을 수행하여 상기 출력 이미지 데이터를 획득하도록 동작 가능한 하나 이상의 프로세싱 유닛을 포함하고,

상기 디스플레이 프로세서는 상기 전자 스크린에 대한 상기 입력 이미지 데이터를 제공하는 하나 이상의 프로세서들과 상기 전자 스크린에 대한 상기 출력 이미지 데이터를 저장하기 위해 이용되는 프레임 버퍼 사이에 인터페이스를 제공하도록 동작 가능한, 무선 디바이스용 디스플레이 프로세서.

청구항 2.

제 1 항에 있어서,

상기 전자 스크린은 액정 디스플레이 (LCD) 스크린인, 무선 디바이스용 디스플레이 프로세서.

청구항 3.

제 1 항에 있어서,

상기 하나 이상의 프로세싱 유닛은 상기 입력 이미지 데이터를 블록으로 프로세싱하도록 동작가능한, 무선 디바이스용 디스플레이 프로세서.

청구항 4.

제 1 항에 있어서,

상기 인터페이스 유닛은 상기 출력 이미지 데이터를 라인으로 제공하도록 동작 가능한, 무선 디바이스용 디스플레이 프로세서.

청구항 5.

제 1 항에 있어서,

상기 하나 이상의 프로세싱 유닛은 상기 입력 이미지 데이터 블록의 뒤집기, 회전, 또는 뒤집기 및 회전을 수행하도록 동작 가능한, 무선 디바이스용 디스플레이 프로세서.

청구항 6.

제 1 항에 있어서,

상기 입력 이미지 데이터는 휘도 성분 (component) 과, 레드 및 블루 색도 성분을 가지는 YCrCb 포맷으로 존재하고,

상기 하나 이상의 프로세싱 유닛은 상기 레드 및 블루 색도 성분들에 대한 상기 입력 이미지 데이터의 업샘플링을 수행하도록 동작 가능한, 무선 디바이스용 디스플레이 프로세서.

청구항 7.

제 1 항에 있어서,

상기 하나 이상의 프로세싱 유닛은 상기 입력 이미지 데이터용 제 1 비디오 포맷으로부터 상기 출력 이미지 데이터용 제 2 비디오 포맷으로 변환을 수행하도록 동작 가능한, 무선 디바이스용 디스플레이 프로세서.

청구항 8.

제 7 항에 있어서,

상기 제 1 비디오 포맷은 휘도 성분과, 레드 및 블루 색도를 가지는 YCrCb 포맷이고,

상기 제 2 비디오 포맷은 레드, 그린, 및 블루 색 성분을 가지는 RGB 포맷인, 무선 디바이스용 디스플레이 프로세서.

청구항 9.

제 1 항에 있어서,

상기 하나 이상의 프로세싱 유닛은 사이즈상 이미지를 크게 또는 작게 스케일링하기 위해 상기 입력 이미지 데이터를 프로세싱하도록 동작 가능한, 무선 디바이스용 디스플레이 프로세서.

청구항 10.

제 1 항에 있어서,

상기 하나 이상의 프로세싱 유닛은 제 1 및 제 2 이미지에 대한 입력 이미지 데이터를 결합하여 합성 이미지에 대한 출력 이미지 데이터를 획득하도록 동작 가능한, 무선 디바이스용 디스플레이 프로세서.

청구항 11.

제 10 항에 있어서,

상기 제 1 이미지는 비디오 또는 그래픽 이미지이고 상기 제 2 이미지는 텍스트에 대한 것인, 무선 디바이스용 디스플레이 프로세서.

청구항 12.

제 10 항에 있어서,

상기 제 1 및 제 2 이미지 데이터에 대한 상기 입력 이미지 데이터를 저장하도록 동작 가능한 하나 이상의 버퍼를 더 포함하는, 무선 디바이스용 디스플레이 프로세서.

청구항 13.

제 1 항에 있어서,

상기 하나 이상의 프로세싱 유닛은 2 개의 이미지의 알파-블렌딩 (alpha-blend) 을 수행하여 합성 이미지를 획득하도록 동작 가능한, 무선 디바이스용 디스플레이 프로세서.

청구항 14.

제 1 항에 있어서,

상기 하나 이상의 프로세싱 유닛은 제 1 계수로 제 1 이미지에 대한 입력 이미지 데이터를 스케일링하고, 제 2 계수로 제 2 이미지에 대한 입력 이미지 데이터를 스케일링하며, 상기 제 1 이미지에 대해 스케일링된 입력 이미지 데이터와 상기 제 2 이미지에 대해 스케일링된 입력 이미지 데이터를 결합하여 합성 이미지 데이터에 대한 상기 출력 이미지 데이터를 획득하는, 무선 디바이스용 디스플레이 프로세서.

청구항 15.

제 1 항에 있어서,

상기 하나 이상의 프로세싱 유닛은 제 2 이미지 위에 제 1 이미지를 중첩시키기 위해 투명성을 수행하도록 동작 가능한, 무선 디바이스용 디스플레이 프로세서.

청구항 16.

제 1 항에 있어서,

상기 하나 이상의 프로세싱 유닛은 하나 이상의 픽셀 위치 각각에 대해,

제 1 이미지에서의 상기 픽셀 위치에 대한 제 1 픽셀값이 투명성 값과 일치하는지 여부를 결정하며,

상기 제 1 픽셀값이 상기 투명성 값과 일치하는 경우, 제 2 이미지에서의 상기 픽셀 위치에 대한 제 2 픽셀값을 상기 픽셀 위치에 대한 출력 픽셀값으로서 제공하고,

그렇지 않은 경우, 상기 제 1 픽셀값을 상기 출력 픽셀값으로서 제공하도록 동작하며,

상기 제 1 및 제 2 픽셀값은 상기 입력 이미지 데이터로부터인 것이고, 상기 출력 픽셀값은 상기 출력 이미지 데이터에 대한 것인, 무선 디바이스용 디스플레이 프로세서.

청구항 17.

제 16 항에 있어서,

상기 제 2 픽셀값은 선택된 색에 대한 소정의 값인, 무선 디바이스용 디스플레이 프로세서.

청구항 18.

제 1 항에 있어서,

명령을 수신하고, 상기 수신된 명령에 따라 상기 하나 이상의 프로세싱 유닛이 상기 입력 이미지 데이터를 프로세싱하게끔 지시하도록 동작 가능한 프로그래머블 명령 프로세서를 더 포함하는, 무선 디바이스용 디스플레이 프로세서.

청구항 19.

제 18 항에 있어서,

상기 명령은 리스트로 수신되고, 상기 명령 프로세서는 종료 조건이 이루어질때까지 상기 리스트에서의 상기 명령을 실행하도록 동작 가능한, 무선 디바이스용 디스플레이 프로세서.

청구항 20.

제 1 항에 있어서,

상기 전자 스크린상에 인위적 영상을 유발하는 것을 방지하기 위해 상기 인터페이스 유닛으로부터 상기 출력 이미지 데이터를 제공하는 타이밍을 결정하도록 동작 가능한 동기화 유닛을 더 포함하는, 무선 디바이스용 디스플레이 프로세서.

청구항 21.

제 20 항에 있어서,

상기 동기화 유닛은 상기 전자 스크린에 대한 각각의 리프레시 사이클을 나타내는 수직 동기화 신호에 기초하여 상기 타이밍을 결정하도록 동작 가능한, 무선 디바이스용 디스플레이 프로세서.

청구항 22.

제 20 항에 있어서,

상기 출력 이미지 데이터는 프레임 버퍼에 순차적으로 저장되고, 상기 프레임 버퍼에 저장된 데이터는 판독 포인터에 기초하여 검색 (retrieve) 되어 상기 전자 스크린상에 디스플레이되며, 상기 판독 포인터를 패스 업 (pass up) 하지 않도록 상기 출력 이미지 데이터가 상기 인터페이스 유닛으로부터 제공되는, 무선 디바이스용 디스플레이 프로세서.

청구항 23.

제 1 항에 있어서,

상기 전자 스크린에 대한 프레임 버퍼가 상기 출력 이미지 데이터를 받아들일 준비가 될때까지 상기 출력 이미지 데이터를 저장하도록 동작 가능한 디바이스 버퍼를 더 포함하는, 무선 디바이스용 디스플레이 프로세서.

청구항 24.

제 23 항에 있어서,

상기 디바이스 버퍼는 상기 전자 스크린에 대한 출력 이미지 데이터의 하나 미만의 완성 프레임을 저장하는, 무선 디바이스용 디스플레이 프로세서.

청구항 25.

제 1 항에 있어서,

상기 인터페이스 유닛은 복수의 가능한 출력 포맷중에서 선택된 출력 포맷으로 상기 출력 이미지 데이터를 제공하도록 동작 가능한, 무선 디바이스용 디스플레이 프로세서.

청구항 26.

입력 이미지 데이터를 수신하고 전자 스크린상의 프리젠테이션을 위한 출력 이미지 데이터를 제공하도록 동작 가능한 인터페이스 유닛; 및

상기 입력 이미지 데이터에 대해 하나 이상의 포스트-프로세싱 기능을 수행하여 상기 출력 이미지 데이터를 획득하도록 동작 가능한 하나 이상의 프로세싱 유닛을 포함하고,

상기 인터페이스 유닛 및 상기 하나 이상의 프로세싱 유닛은 상기 전자 스크린에 대한 상기 입력 이미지 데이터를 제공하는 하나 이상의 프로세서와 상기 전자 스크린에 대한 상기 출력 이미지 데이터를 저장하기 위해 이용되는 프레임 버퍼 사이에 인터페이스를 제공하도록 동작 가능한, 무선 디바이스용 집적 회로.

청구항 27.

제 26 항에 있어서,

상기 하나 이상의 프로세싱 유닛은 블록으로 상기 입력 이미지 데이터를 프로세싱하도록 동작 가능하고, 상기 인터페이스 유닛은 라인으로 상기 출력 이미지 데이터를 제공하도록 동작 가능한, 무선 디바이스용 집적 회로.

청구항 28.

제 26 항에 있어서,

상기 하나 이상의 프로세싱 유닛은 영역 뒤집기, 영역 회전, 색 변환, 이미지 사이즈 리스케일링, 알파-블렌딩, 투명성, 텍스트 오버레이, 또는 이들의 조합을 수행하도록 동작 가능한, 무선 디바이스용 집적 회로.

청구항 29.

제 26 항에 있어서,

리스트로 명령을 수신하고 종료 조건이 이루어질때까지 상기 리스트의 상기 명령을 실행하며, 상기 명령에 따라 상기 하나 이상의 프로세싱 유닛이 상기 입력 이미지 데이터를 프로세싱하게끔 지시하도록 동작 가능한 명령 프로세서를 더 포함하는, 무선 디바이스용 집적 회로.

청구항 30.

제 26 항에 있어서,

상기 전자 스크린에 대한 각각의 리프레시 사이클을 나타내는 수직 동기화 신호를 수신하고 상기 전자 스크린상에 인위적 영상을 유발하는 것을 방지하기 위해 상기 인터페이스 유닛으로부터 상기 출력 이미지 데이터를 제공하는 타이밍을 결정하도록 동작 가능한 동기화 유닛을 더 포함하는, 무선 디바이스용 집적 회로.

청구항 31.

전자 스크린에 대한 입력 이미지 데이터를 수신하는 수단;

상기 입력 이미지 데이터에 대해 하나 이상의 포스트-프로세싱을 수행하여 출력 이미지 데이터를 획득하는 수단; 및

상기 전자 스크린상의 프리젠테이션을 위해 상기 출력 이미지 데이터를 제공하는 수단을 포함하고,

상기 입력 이미지 데이터를 수신하는 수단, 상기 하나 이상의 포스트-프로세싱 기능을 수행하는 수단, 상기 출력 이미지 데이터를 제공하는 수단은 상기 전자 스크린에 대한 상기 입력 이미지 데이터를 제공하는 하나 이상의 프로세서와 상기 전자 스크린에 대한 상기 출력 이미지 데이터를 저장하기 위해 이용되는 프레임 버퍼 사이에 인터페이스를 제공하는, 장치.

청구항 32.

제 31 항에 있어서,

상기 하나 이상의 포스트-프로세싱 기능은 영역 뒤집기, 영역 회전, 색 변환, 이미지 사이즈 리스케일링, 알파-블렌딩, 투명성, 텍스트 오버레이, 또는 이들의 조합을 포함하는, 장치.

청구항 33.

제 31 항에 있어서,

상기 전자 스크린에 대한 각각의 리프레시 사이클을 나타내는 수직 동기화 신호를 수신하는 수단; 및

상기 전자 스크린상에 인위적 영상을 유발하는 것을 방지하기 위해, 상기 수직 동기화 신호에 기초하여 상기 출력 이미지 데이터를 제공하는 타이밍을 결정하는 수단을 더 포함하는, 장치.

명세서

이 출원은 2004 년 2 월 24 일 제출되고 발명의 명칭이 "무선 디바이스용 디스플레이 프로세서" 인 미국 가출원 제 60/547,711 호 및 2004 년 5 월 6 일 제출되고 발명의 명칭이 "이동 디스플레이 프로세서 (MDP) HLD 문서" 인 미국 가출원 제 60/569,288 호의 우선권을 주장한다.

배경

I. 분야

본 발명은 일반적으로 회로, 및 더 구체적으로는 디스플레이 프로세서에 관한 것이다.

II. 배경

무선 통신 디바이스 (예를 들어, 셀룰러 폰) 는 음성 및 데이터 통신을 제공하도록 널리 이용된다. 이들 무선 디바이스는 그들의 연산능력 및 메모리 사이즈가 증가함에 따라 증가하는 기능 및 애플리케이션의 개수를 제공한다. 예를 들어, 다수의 무선 디바이스는 정지 이미지 및/또는 동영상을 캡처하고 프로세스하며, 비디오 게임 등을 지원하는 기능들을 가진다.

비디오 및 그래픽은 일반적으로 디지털 신호 프로세서 (DSP), 마이크로 프로세서 등에 의해 제공될 수도 있는 막대한 프로세싱 전력을 요구한다. 이들 프로세서는 미처리 이미지 데이터에 대해 요구된 프로세스 모두를 수행하고 사용자에게 프리젠테이션하기 위해 디스플레이 스크린에 출력 이미지 데이터를 제공한다. 이들 프로세서들은 또한 다른 기능 및 애플리케이션을 지원할 수도 있다. 프로세서에 의해 수행되는 각각의 비디오/그래픽 태스크는 특정 자원량을 소비하고, 이는 다른 기능 및 애플리케이션에 이용가능한 자원량을 감소시킨다.

비디오 및 그래픽 애플리케이션은 또한 통상적으로 외부 메모리로부터 미처리 이미지 데이터를 패치하고 외부 메모리에 프로세싱된 이미지 데이터를 되돌려 기록하기 위해 다수의 메모리 액세스를 요구한다. 메모리 액세스는 외부 버스 인터페이스 (EBI) 를 통해 수행될 수도 있다. 이미지 데이터가 패치되거나 되돌려 기록되는 시간동안, EBI 는 묶이고 다른 프로세스에 의해 및/또는 다른 애플리케이션을 위해 이용될 수 없으며, 이는 매우 바람직하지 않다.

무선 디바이스용 디스플레이 사이즈는 현재 이 분야의 기술 상태에 따라 비교적 작지만, 기술이 개선됨에 따라 증가하기 쉽다. 디스플레이 사이즈가 커지고 색상수 (color depth) 가 증가함에 따라, 비디오 및 그래픽 애플리케이션은 더욱 정교해지고, 더 많은 프로세서 자원을 소비하며, 더 많은 메모리 액세스를 요구하기 쉽다.

따라서 이 분야에서는 특히 디스플레이 사이즈 및/또는 색상수가 증가함에 따라 무선 디바이스에서 비디오 및 그래픽 애플리케이션을 효율적으로 지원하는 기술이 필요하다.

요약

무선 디바이스에서 비디오 및/또는 그래픽 프로세서와 프레임 버퍼간의 인터페이스를 효율적으로 제공할 수 있는 디스플레이 프로세서가 여기서 설명된다. 프레임 버퍼는 전자 스크린 예를 들어, 액정 디스플레이 (LCD) 스크린에 대한 이미지 데이터를 저장한다. 디스플레이 프로세서는 LCD 스크린용 프레임 버퍼에 이미지 데이터를 효율적으로 업데이트할 수 있다. 디스플레이 프로세서는 LCD 스크린에 디스플레이될 이미지 데이터에 대한 포스트-프로세싱을 분주하게 (on-the-fly) 수행할 수 있다. 이는 또한 비디오/그래픽 프로세서가 다른 태스크를 수행하고 디스플레이되지 않는 이미지 데이터에 낭비되는 프로세싱을 방지하도록 한다. 디스플레이 프로세서는 또한 이미지 데이터를 저장하도록 이용되는 메인 메모리에 대한 메모리 액세스 개수를 감소시킨다. 디스플레이 프로세서는 포스트-프로세싱, 구성하기 (composing), 변형하기, 및 복수의 자원으로부터 직교영역의, LCD 스크린으로 전송되는 복합 프레임으로의 변환을 수행할 수도 있다.

일 실시형태에서, 디스플레이 프로세서는 인터페이스 유닛, 명령 프로세서, 동기화 유닛, 하나 이상의 프로세싱 유닛, 및 디바이스 버퍼를 포함한다. 인터페이스 유닛은 입력 이미지 데이터를 수신하고 (예를 들어, 메인 메모리로부터) 프레임 버퍼에 출력 이미지 데이터를 제공한다. 명령 프로세서는 명령을 수신하고 (예를 들어, 무선 디바이스내의 제어기로부터) 프로세싱 유닛(들) 의 동작을 지시한다. 명령은 리스트로 주어질 수도 있고, 종료 조건 (예를 들어, STOP 명령) 이 이루어질 때까지 명령 프로세서는 리스트의 명령을 실행할 수도 있다. 동기화 유닛은 LCD 스크린상의 인위적 영상 (visual artifact) 을 방지하기 위해 프레임 버퍼에 대한 판독 포인터의 위치를 결정하고 프레임 버퍼에 출력 이미지 데이터의 기록을 제어한다. 프로세싱 유닛(들) 은 영역 뒤집기, 영역 회전, 하나의 비디오 포맷 (예를 들어, YCrCb) 으로부터 또 다른 비디오 포맷 (예를 들어, RGB) 로의 색상 변환, 업/다운 이미지 사이즈 리스케일링, 알파-블렌딩 (alpha-blending), 투명성 (transparency), 텍스트 오버레이 (text overlay) 등과 같은 다양한 포스트-프로세싱 기능을 수행할 수도 있다.

본 발명의 다양한 양상 및 실시형태를 이하 더 상세히 설명한다.

도면의 간단한 설명

본 발명의 특징 및 본질은 도면과 함께 이해했을 때 아래에 진술된 상세한 설명으로부터 더 명백해질 것이며, 도면에서, 동일한 참조 부호를 명세서 전반에 걸쳐 대응하여 식별된다.

도 1 은 무선 디바이스의 블록도를 도시한다.

도 2 는 디스플레이 프로세서에 의해 생성된 예시적인 디스플레이를 도시한다.

도 3 은 디스플레이 프로세서의 블록도를 도시한다.

도 4a 및 4b 는 2 개의 시간의 프레임 버퍼의 콘텐츠를 도시한다.

도 5 는 16 개의 상이한 회전 및 뒤집기 조합에 대한 8 개의 배향을 도시한다.

도 6 은 이미지의 90° 회전을 도시한다.

도 7a 는 4:2:0 YCrCb 포맷에 대한 8×8 픽셀 블록을 도시한다.

도 7b 는 쌍선형 (bilinear) 필터를 이용한 색도 (chrominance) 업샘플링을 도시한다.

도 8a, 8b, 및 8c 는 각각 1/2, 3/8 및 5/8 다운스케일링을 도시한다.

도 9a 및 9b 는 행과 열 각각에 대한 바이큐빅 내삽 (bicubic interpolation) 을 도시한다.

상세한 설명

용어 "예시적인" 은 여기서 "예, 예시 또는 예증으로서 제공되는" 의 의미로 사용된다. 임의의 실시형태 또는 "예시적인" 으로서 설명된 설계는 반드시 다른 실시형태 또는 설계보다 선호되거나 유리한 것으로 해석될 필요는 없다.

도 1 은 무선 통신 시스템에서의 무선 디바이스 (100) 의 블록도를 도시한다. 무선 디바이스 (100) 는 셀룰러 폰, 단말기, 핸드셋 (handset), 다중-단면 개인 휴대 정보 단말기 (PDA), 또는 다른 일부 장치일 수도 있다. 무선 통신 시스템은 코드 분할 다중 접속 (CDMA: Code Division Multiple Access) 시스템, 전 지구적 이동 통신 (GSM: Global System for Mobile Communication) 시스템 등일 수도 있다. 무선 디바이스 (100) 는 수신 경로 및 송신 경로를 통해 쌍방향 통신을 제공할 수 있다.

수신경로에 대해, 기지국에 의해 송신된 신호는 안테나 (112) 에 의해 수신되고, 듀플렉서 (D: 114: duplexer) 를 통해 라우팅되어 수신기 유닛 (RCVR: 116) 에 제공된다. 수신기 유닛 (116) 은 수신 신호를 조절하고 디지털화하여 추가적인 프로세싱을 위해 디지털부 (120) 에 입력 샘플을 제공한다. 송신 경로에 대해, 송신기 유닛 (TMTR: 118) 은 디지털부 (120) 로부터 송신될 데이터를 수신하고 이 데이터를 프로세싱하고 조절하여 듀플렉서 (114) 를 통해 라우팅되고 안테나 (112) 를 통해 기지국으로 송신되는 변조 신호를 발생시킨다.

디지털부 (120) 는 예를 들어, 모뎀 프로세서 (122), 비디오 프로세서 (124), 그래픽 프로세서 (126), 애플리케이션 프로세서 (128), 제어기 (130), 디스플레이 프로세서 (140), 외부 버스 인터페이스 (EBI: 142 및 144), 및 이동 디지털 디스플레이 인터페이스 (MDDI) 호스트 (146) 와 같은 다양한 프로세싱 및 인터페이스 유닛을 포함한다. 모뎀 프로세서 (122) 는 데이터 송신 및 수신 (예를 들어, 인코딩, 변조, 복조, 디코딩 등) 을 위한 프로세싱을 수행한다. 비디오 프로세서 (124) 는 캡 코더, 비디오 플레이백, 비디오 컨퍼런싱 등과 같은 비디오 애플리케이션에 대한 비디오 콘텐츠 (예를 들어, 정지 이미지, 동영상, 이동 텍스트 등) 에 대한 프로세싱을 수행한다. 그래픽 프로세서 (126) 는 비디오 게임, 3-D 아바타 등과 같은 그래픽 애플리케이션에 대한 그래픽 (예를 들어, 2-차원 (2D) 모델, 3D 모델 등) 에 대한 프로세싱을 수행한다. 애플리케이션 프로세서 (128) 는 예를 들어, 다중-웨이 콜 (multi-way calls), 웹 브라우징, 폰 다이얼러 (phone dialer) 애플리케이션, 미디어 플레이어, 게임, 사용자 인터페이스 등과 같은 다양한 애플리케이션에 대한 프로세싱을 수행한다. 디스플레이 프로세서 (140) 는 LCD 유닛 (180) 상에서 비디오, 그래픽, 텍스트 등의 디스플레이를 용이하게 하는 특정 포스트-프로세싱 태스크를 수행한다. LCD 유닛 (180) 은 예를 들어, 박막 트랜지스터 (TFT: thin film transistor), 유기 전기 발광 다이오드 (OLED: organic light emitting diode), 브라운관 (CRT: cathode ray tube) 등과 같은 임의의 전자 디스플레이 유형일 수도 있다. 제어기 (130) 는 디지털부 (120) 내의 다양한 프로세싱 및 인터페이스 유닛의 동작을 지시한다. 디지털부 (120) 내의 다양한 유닛들은 하나 이상의 버스 (132) 를 통해 통신할 수도 있다.

EBI (142 및 144) 는 또한 버스 (132) 에 결합된다. EBI (142) 는 디지털부 (120) 와 랜덤 액세스 메모리 (RAM), 정적 RAM (SRAM), 동적 RAM (DRAM), 동기식 DRAM (SDRAM) 등일 수도 있는 휘발성 메인 메모리 (152) 사이에 데이터의 전송을 용이하게 한다. EBI (144) 는 디지털부 (120), 비휘발성 메모리 (154: 예를 들어, NAND 플래시 메모리), 및 LCD 유닛 (180) 간의 데이터의 전송을 용이하게 한다. EBI (144) 는 또한 전용 포트를 통해 디스플레이 프로세서 (140) 에 직접 결합될 수도 있다 (도 1 에 도시된 바와 같음). 디스플레이 프로세서 (140) 가 데이터 전송을 위한 전용 포트를 완전히 차지

하는 경우에도, 다른 프로세서들은 데이터의 교환을 위해 EBI (144) 를 이용할 수도 있다. MDDI 호스트 (146) 는 디지털부와 LCD 유닛 (180) 간의 효율적인 고속 직렬 인터페이스 및 컴퓨터, 주변기기, 및 소비자 가전 제품간의 효율적인 데이터 인터페이싱을 제공하는 IEEE 1394 파이어와이어와 유사한 방식으로 기능한다. MDDI 호스트 (146) 는 예를 들어, 직-병렬 변환, 네트워크 제어 등과 같은 다양한 기능들을 수행할 수도 있다.

디지털부 (120) 는 하나 이상의 DSP, 마이크로 프로세서, 축소 명령 세트 컴퓨터 (RISC: reduced instruction set computer) 등으로 구현될 수도 있다. 디지털부 (120) 는 또한 하나 이상의 주문형 집적회로 (ASIC: application specific integrated circuit) 또는 일부 다른 형태의 집적 회로 (IC) 상에서 구성될 수도 있다.

도 1 에 도시된 실시형태에 대해, LCD 유닛 (180) 은 LCD 제어기 (182), MDDI 클라이언트 (184), 프레임 버퍼 (186) 및 LCD 스크린 (188) 을 포함한다. LCD 제어기 (182) 는 EBI (144) 와 인터페이싱하고 디지털부 (120) 로부터 프레임 버퍼 (186) 로 이미지의 전송을 용이하게 한다. MDDI 클라이언트 (184) 는 MDDI 호스트 (146) 와 인터페이싱하거나 대신 프레임 버퍼 (186) 에 이미지 데이터를 효율적으로 전송하도록 이용될 수도 있다. 또 다른 실시형태에서, MDDI 호스트 (146) 및 MDDI 클라이언트 (184) 는 무선 디바이스 (100) 및 LCD 유닛 (180) 으로부터 각각 생략될 수도 있다. 프레임 버퍼 (186) 는 LCD 스크린 (180) 에서 디스플레이될 이미지 데이터의 프레임을 저장한다. 다음의 설명에서, "이미지 데이터" 는 그림 구성요소 (픽셀) 데이터와 교환적으로 사용되고 LCD 스크린 (188) 상의 프리젠테이션에 적합한 데이터이다.

도 2 는 디스플레이 프로세서 (140) 에 의해 생성될 수도 있는 예시적인 디스플레이 (200) 를 도시한다. 디스플레이 (200) 는 복수의 영역으로 분할된다. 각각의 영역은 상이한 데이터 자원 및 상이한 업데이트 레이트와 관련될 수도 있다. 도 2 에 도시된 실시형태에 대해, 디스플레이 (200) 는 폰 상태 영역 (212), 활성 애플리케이션 영역 (214), 비디오/그래픽 영역 (216), 텍스트 오버레이 영역 (218), 및 애플리케이션 제어 영역 (220) 을 포함한다.

폰 상태 영역 (212) 은 무선 디바이스 (100) 의 상태를 표시하는 다양한 예고장치 (annunciators; 또는 아이콘) 를 도시할 수도 있다. 이러한 예고장치는 예를 들어, 모드 표시자 (예를 들어, 디지털 (D) 또는 아날로그), 로밍 표시자 (예를 들어, 로밍 (R) 또는 네트워크), 신호 강도 표시자, "사용중" 표시자, 음성메일 표시자, 그날의 시간 (the time-of-the-day) 표시자, 배터리 레벨 표시자 등을 포함할 수도 있다. 영역 (212) 은 애플리케이션 프로세서 (128) 에 의해, 예를 들어, 초당 약 한번의 레이트로 업데이트될 수도 있다. 일부 예고장치는 정적이고 인에이블 또는 디스에이블중 하나가 될 수도 있다. 다른 예고장치들 (예를 들어, 그날의 시간 및 신호강도 표시자) 은 주기적으로 업데이트될 수도 있다. 디스플레이 프로세서 (140) 는 영역 (212) 에 대한 이미지 데이터를 LCD 유닛 (180) 으로 간단히 전송할 수도 있다.

활성 애플리케이션 영역 (214) 은 무선 디바이스 (150) 상에서 동작하고 있는 애플리케이션에 대한 다양한 제어를 도시할 수도 있다. 이들 제어는 예를 들어, 비디오 제어, 볼륨 제어, 테이프 카운터 등일 수도 있다. 영역 (214) 에 대한 업데이트 레이트는 상당히 느릴 수도 있다. 디스플레이 프로세서 (140) 는 영역 (214) 에 대한 이미지 데이터를 LCD 유닛 (180) 으로 간단히 전송시킬 수도 있다.

비디오/그래픽 영역 (216) 은 정지 이미지, 비디오, 그래픽, 텍스트 등을 나타낼 수도 있다. 디스플레이 프로세서 (140) 는 후술하는 바와 같이, 비디오 및 그래픽에 대한 이미지 데이터에 대해 포스트-프로세싱을 수행할 수도 있다. 영역 (216) 은 지정된 레이트 (예를 들어, 약 5 Hertz 내지 30 Hertz) 로 업데이트 될 수도 있다. 영역 (216) 은 디스플레이 (200: 도 2 에 도시된 바와 같음) 의 부분일 수도 있고 또는 디스플레이 전체를 차지할 수도 있다. 비디오, 그래픽, 및 텍스트는 또한 90° 로 회전될 수도 있고 수평 (수직 대신) 방향으로 디스플레이 될 수도 있다. 영역 (216) 은 예를 들어, 사용자가 섬네일 사진 (photograph thumbnails) 을 통해 탐색하는 경우에, 복수의 이미지를 나타낼 수도 있다. 영역 (216) 은 또한 예를 들어, 비디오들중 하나가 로컬 종단이 될 수도 있는 하나 이상의 다른 위치에서의 다른 사용자와 원격전화회의 콜 (teleconference call) 을 위해 복수의 비디오들을 나타낼 수도 있다. 디스플레이 프로세서 (140) 는 LCD (180) 로의 전송에 앞서 각각의 이미지/비디오에 대한 회전을 수행한다 (필요한 경우). 디스플레이 프로세서 (140) 는 또한 사용자가 거울을 보는데와 동일한 친근한 효과를 경험하도록 원격전화회의 콜에서 각각의 비디오에 대해 좌-우 뒤집기를 수행할 수도 있다.

텍스트 오버레이 영역 (218) 은 비디오/그래픽을 통해 오버레이된 텍스트를 나타낼 수도 있다. 오버레이 텍스트는 스크롤되거나 정지할 수도 있고 주식, 자막방송 (closed-caption) 텍스트 등과 같은 다양한 유형의 정보에 대한 것일 수도 있다. 텍스트를 스크롤링하는 것은 배경 비디오/그래픽과 동일한 레이트로 업데이트될 수도 있다.

애플리케이션 제어 영역 (220) 은 애플리케이션에 대한 다양한 제어를 나타낼 수도 있다. 이들 제어들은 2-D 아이콘, 3-D 아바타 등의 형태일 수도 있다. 전체 디스플레이 콘텐츠가 변하지 않는다면 아이콘은 일단 구성되어 정지한 채로 유지할 수도 있다. 아바타는 대화식 일 수 있고, 디스플레이 (200) 내에서 다양한 영역을 움직일 수도 있고, 동작 (예를 들어, 저글

링 (juggle)) 중 일 수도 있으며, 기타 상태일 수도 있다. 디스플레이 프로세서 (140) 는 LCD 유닛 (180) 으로 전향할 배경, 아이콘, 및 아바타에 대한 이미지 데이터를 구성할 수도 있다. 아바타가 활성화되는 경우, 디스플레이 프로세서의 투명성 기능을 이용하여, 디스플레이 프로세서 (140) 는 또한 충분한 레이트로 (예를 들어, 초당 약 20 Hertz) 로 배경의 최상위에 아바타를 업데이트할 수도 있다.

도 2 는 예시적인 디스플레이를 도시한다. 일반적으로, 디스플레이는 임의의 영역 개수를 포함할 수도 있고, 각각의 영역은 임의의 정보 유형을 나타낼 수도 있다. 상이한 정보 유형은 디스플레이 프로세서 (140) 에 의해 상이한 방식으로 프로세싱될 수도 있다.

도 1 을 다시 참조하면, 프로세서 (124 내지 128) 는 LCD 스크린 (188) 상의 디스플레이를 위한 이미지 데이터를 제공할 수도 있다. 이들 프로세서는 애플리케이션 디바이스 버퍼로서 이용될 수도 있는 메인 메모리 (152) 에 그들의 이미지 데이터를 저장할 수도 있다. 디스플레이 프로세서 (140) 는 메인 메모리 (152) 와 프레임 버퍼 (186) 사이에 위치하고, 메인 메모리 (152) 로부터 LCD 유닛 (180) 으로 이미지 데이터를 효율적으로 전송시킬 수 있다. 공통 디스플레이 프로세서 (140) 는 프로세서 (124 내지 128) 를 효율적으로 지원하고 리던던트 회로 구성요소 (redundant circuitry) 를 감소시킬 수 있다. 디스플레이 프로세서 (140) 는 메인 메모리 (152) 로부터 이미지 데이터를 검색하고, 프레임 버퍼 (186) 에 이미지 데이터를 기록하여 이미지 전송을 수행한다. LCD 스크린 (188) 에 대한 콘텐츠가 변경되는 경우마다 프레임 버퍼 (186) 는 디스플레이 프로세서 (140) 에 의해 새로운 이미지 데이터로 "업데이트" 된다. LCD 스크린 (188) 은 프레임 버퍼 (186) 로부터 주기적으로 이미지 데이터로 "리프레시 (refresh)" 된다 (예를 들어, 60 Hertz 내지 75 Hertz).

디스플레이 프로세서 (140) 는 LCD 유닛 (180) 으로 이미지 데이터를 전송시키는 동안 이미지 데이터에 대해 하나의 세트의 포스트-프로세싱을 분주하게 수행할 수도 있다. 이미지 데이터가 소정의 태스크에 대해 복수의 회수로 메모리로부터 판독되거나, 메모리에 기록되지 않도록, 이들 포스트-프로세싱 기능들은 프로세서 (124 내지 128) 상의 프로세싱 부담을 감소시키고 (1), 메인 메모리 (152) 의 대역폭 효율성을 개선시키도록 (2) 선택된다. 일 실시형태에서, 디스플레이 프로세서 (140) 는 다음의 기능들을 지원한다.

- 프레임 업데이트 동기화 - LCD 스크린 (188) 상의 인위적 영상을 유발하는 것을 방지하는 방식으로 프레임 버퍼 (186) 를 업데이트.
- 이미지 변형 - 회전 (90°, 180°, 270°), 뒤집기/반사 (좌/우 및/또는 최상/바닥), 및 스케일링;
- 프레임 구성 - 이미지 복사, 투명성으로 복사, 및 알파-블렌딩;
- 포맷 변환 - 색상수 (업/다운 샘플링), 색공간 (color space: YCrCb & RBG) 및 색포맷 (color format: 565, 666, 4:2:0 등);
- 스크립트 프로세싱 - 스크립트로서 저장된 명령들을 실행.

이들 기능들은 이하 더 상세히 설명한다. 디스플레이 프로세서 (140) 는 또한 더 적은, 상이한, 및/또는 추가적인 기능들을 수행하도록 설계될 수도 있다.

도 3 은 디스플레이 프로세서 (140) 의 일 실시형태의 블록도를 도시한다. 버스 인터페이스 유닛 (312) 은 디스플레이 프로세서 (140) 에 대해 데이터 교환을 취급한다. 버스 인터페이스 (312) 는 버스 (132) 의 설계에 기초하여 (예를 들어, 각각의 버스트 (burst) 가 4 개의 클럭 사이클에서 32 비트 버스를 통해 32 비트를 제공하는 4 개의 비트 버스트로) 효율적인 방식으로 메인 메모리 (152) 로부터 입력 이미지 데이터를 수신한다. 페치 유닛 (314) 은 데이터 판독을 위해 버스 인터페이스 유닛 (312) 에 대한 주소를 생성한다. 입력 이미지 데이터는 프로세싱되고 LCD 유닛 (180) 으로 포워딩하기 위해 EBI (144) 또는 MDDI 호스트 (146) 중 하나로 버스 인터페이스를 통해 전송된다. 버스 인터페이스 유닛 (312) 은 선택된 포맷으로 (예를 들어, 단일 비트 워드 전송에 봉인된 단일의 픽셀 시리즈로서) 출력 이미지 데이터를 제공한다.

명령 프로세서 (320) 는 명령/코멘트 (예를 들어, 제어기 (130) 로부터) 를 수신하고 디스플레이 프로세서 (140) 내의 다양한 유닛의 동작을 지시한다. 이 명령들은 LCD 스크린의 전부 또는 일부에 대해 이미지 데이터를 업데이트하기 위해 디스플레이 프로세서 (140) 에 의해 수행될 기능들을 설명한다. 명령들은 스크립트 (또는 리스트) 에 제공될 수도 있다. 이러한 경우에, 명령 프로세서 (320) 는 종료 조건 (예를 들어, STOP, END, 또는 HALT 명령) 이 이루어질때까지 스크립트의 명령들을 프로세싱한다. 이 시기에, 명령 프로세서 (320) 는 스크립트가 완료되었고 유휴 (idle) 로 되었으며 다음 스크립트

를 기다린다는 것을 제어기 (130) 에 알리기 위해 상태 플래그 (status flag) 를 설정할 수도 있다. 스크립트는 메인 메모리 (152) 또는 비휘발성 메모리 (154) 에 저장될 수도 있고, 명령 프로세서 (320) 는 실행될 다음 스크립트에 대한 포인터가 제공될 수도 있다.

레지스터 (324) 는 디스플레이 프로세서 (140) 내의 다양한 프로세싱에 대한 파라미터들을 저장하고 제어기 (130) 가 디스플레이 프로세서를 모니터링하고 제어하도록 한다. 레지스터 (324) 의 파라미터들은 스크립트 및/또는 명령 프로세서 (320) 에 의해 설정될 수도 있고 모니터링 될 수도 있다 (예를 들어, 시험 및 디버그 (debug) 목적).

동기화 (sync) 유닛 (326) 은 프레임 버퍼 (186) 에 대한 판독 포인터 및 기록 포인터를 추적하고 LCD 스크린상에서 "티어링 (tearing)" 을 유발하지 않고 프레임 버퍼 (186) 에 새로운 프레임에 대한 이미지 데이터를 기록하는 것이 안전한지 여부를 결정한다. LCD 스크린은 통상적으로 프레임 버퍼 (186) 로부터의 이미지 데이터로, 한번에 한 라인씩 순차적으로 리프레시된다. 판독 포인터는 LCD 스크린을 리프레시하기 위해 프레임 버퍼 (186) 로부터 검색 (retrieve) 하는 현재 프레임에 대한 이미지 데이터의 다음 라인을 나타낸다. 판독 포인터는 새로운 프레임에 대한 이미지 데이터가 저장되는 프레임 버퍼 (186) 에서 라인을 표시한다. LCD 스크린 (188) 상의 인위적 영상의 유발을 방지하기 위해, 기록 포인터는 판독 포인터를 지연시켜야 한다. 새로운 프레임에 대한 이미지 데이터가 판독 포인터 이후 프레임 버퍼 (186) 에 기록되는 경우, LCD 스크린 (188) 은 현재 프레임에 대한 이미지 데이터 직후의 새로운 프레임에 대한 이미지 데이터로 리프레시될 것이다. 보통 티어링이라 칭하는 바람직하지 않은 인위물이 LCD 스크린에 나타난다. 동기화 유닛 (326) 은 LCD 유닛 (180) 의 타이밍에 커스 락킹 (coarse lock) 을 유지하여 기록 포인터가 판독 포인터를 패스하지 않도록 보장한다.

일 실시형태에서, 디스플레이 프로세서 (140) 는 블록들의 이미지 데이터상에서 동작한다. H 가 프레임의 라인 개수이고 W 가 라인당 픽셀 개수인 $H \times W$ 차원 프레임은 2-D 블록 배열로 분할될 수도 있다. 일반적으로, 블록은 임의의 차원이 되도록 규정될 수도 있고 스퀘어 (square) 일 필요는 없다. 블록 사이즈는 다양한 요소들 예를 들어, 메모리 액세스가 버스 (132) 를 통해 효율적으로 수행되는 방식, 디스플레이 프로세서 (140) 내의 유닛의 프로세싱 기능, 프레임 사이즈 등에 기초하여 선택될 수도 있다. N 이 2 의 제곱인승, $N \times N$ 차원의 스퀘어 블록은 양호한 성능을 제공할 수도 있다. 예를 들어, 디스플레이 프로세서 (140) 는 16×16 블록, 8×8 블록 등에서 동작할 수도 있다. 16×16 블록 사이즈는 버스 (132) 가 최적화되어 각각의 메모리 액세스에 16 바이트 데이터를 제공하는 경우에, 유리할 수도 있다. 명료함을 위해, 이하의 설명의 대부분은 8×8 블록 사이즈에 대한 것이다.

뒤집기 및 회전 유닛 (330) 은 버스 인터페이스 유닛 (312) 으로부터 입력 이미지 데이터를 수신하고, 명령 프로세서 (320) 에 의해 지시되는 바와 같이 입력 이미지 데이터 상에서 뒤집기 및/또는 회전을 수행하여, 입력 버퍼 (332) 에 그 출력 데이터를 저장한다. 이미지는 복수의 스트립으로 분할될 수도 있고, 각각의 스트립은 이미지의 전체폭에 이를 수도 있고 복수의 블록들을 포함한다. 유닛 (330) 은 하나 이상의 스트립에 대해 소정의 스트립내의 하나 이상의 블록상에서 뒤집기 및/또는 회전등을 수행할 수도 있다. 입력 버퍼 (332) 는 메인 메모리 (152) 로부터 폐치되고 뒤집기 및 회전 유닛 (330) 에 의해 프로세싱된 이미지 데이터 블록을 저장한다. 입력 버퍼 (332) 의 이미지 데이터 블록은 LCD 스크린에 대해 정확한 배향이다.

색 변환 및 스케일링 유닛 (334) 은 입력 버퍼 (332) 로부터 데이터를 수신하고 필요한 경우 입력 비디오 포맷으로부터 출력 비디오 포맷으로 데이터를 변환하고, 제 1 버퍼 (336) 또는 제 2 버퍼 (338) 중 하나에 픽셀 데이터를 저장한다. 입력 비디오 포맷은 예를 들어, 휘도 및 색도 (YCrCb) 포맷일 수도 있고, 출력 비디오 포맷은 예를 들어, 레드, 그린, 및 블루 (RGB) 포맷일 수도 있다. 유닛 (334) 은 또한 버퍼 (336 또는 338) 에 저장하기에 앞서 이미지를 사이즈상 위 아래로 스케일링할 수도 있다.

제 1 버퍼 (336) 는 주요 이미지에 대한 데이터 블록을 저장한다. 이러한 데이터 블록은 디스플레이 프로세서 (140) 가 이미지의 에지를 프로세싱하는 경우 일반 $N \times N$ 블록 사이즈 또는 더 작은 사이즈 블록일 수도 있다. 제 2 버퍼 (338) 는 제 2 이미지에 대한 데이터 블록을 저장한다. 버퍼 (336 및 338) 에 저장된 2 개의 데이터 블록은 합성 이미지에 대한 출력 블록을 형성하기 위해 믹싱되고 결합될 수도 있다. 믹싱 또는 텍스트 오버레이가 수행되지 않는 경우, 제 1 버퍼 (336) 는 LCD 스크린에 대한 데이터 블록을 저장하고 제 2 버퍼는 유향이 된다. 어떠한 경우에도, 제 1 버퍼 (336) 는 디바이스 버퍼 (342) 에 블록에 대한 룬이 있을 때까지 데이터 블록을 홀딩한다.

믹스, 블렌드, 및 크롭 유닛 (340; mix, blend & crop unit) 은 버퍼 (336 및 338) 로부터 데이터 블록을 수신하고, 적용가능한 경우 데이터 블록을 결합하고, 디바이스 버퍼 (342) 에 출력 데이터 블록을 제공한다. 유닛 (340) 은 입력 데이터 블록의 크롭을 수행할 수도 있다. 유닛 (340) 은 또한 디바이스 버퍼 (342) 로 출력하기에 앞서 데이터 블록에 대해 알파-블렌딩 또는 투명성 변형을 수행할 수도 있다.

디바이스 버퍼 (342) 는 "가상" 프레임 버퍼로서 작용하고, 유닛 (340) 으로부터 픽셀 데이터 블록을 받아들이고, LCD 유닛 (180) 이 기대하는 행 포맷으로 픽셀 데이터를 제공한다. 디바이스 버퍼 (342) 는 프레임 버퍼에 픽셀 데이터 라인을 효율적으로 제공하기 위해 프레임 버퍼 (186) 와 동일한 폭을 가질 수도 있다. 일 실시형태에서, 디바이스 버퍼 (342) 는 $H \times W$ 프레임 사이즈의 일부만을 커버하는 2 개의 데이터 블록 행을 저장한다. 데이터 블록의 하나의 행으로부터의 라인은 디바이스 버퍼 (342) 로부터 검색되어 프레임 버퍼 (186) 에 기록될 수도 있고, 또 다른 행에 대한 새로운 데이터 블록은 디바이스 버퍼 (342) 에 기록된다. 또 다른 실시형태에서, 디바이스 버퍼 (342) 는 하나의 데이터 블록 행을 저장한다. 라인이 디바이스 버퍼 (342) 로부터 검색되고 프레임 버퍼 (186) 에 기록되는 경우마다, 충분한 메모리 공간이 하나 이상의 새로운 데이터 블록을 저장하기 위해 프리업 (freed up) 될 수도 있다. 이러한 실시형태에 대해, 각각의 데이터 블록은 디바이스 버퍼 (342) 내의 인접하지 않은 위치에 저장될 수도 있고, 디바이스 버퍼 (342) 에 기록되거나 디바이스 버퍼 (342) 로부터 검색되는 각각의 블록에 대해 적합한 메모리 주소가 생성된다. 그러나 또 다른 실시형태에서, 디바이스 버퍼 (342) 는 도 2 에 도시된 바와 같은 모든 영역에 대해 전체 프레임 예를 들어, 비디오, 그래픽, 텍스트 등에 대한 이미지 데이터를 저장한다. 일반적으로, 디바이스 버퍼 (342) 는 모든 또는 일부의 프레임에 대한 이미지 데이터를 저장하고, 다양한 방식으로 이미지 데이터를 저장할 수도 있다. 디바이스 버퍼 (342) 는 그 실제 사이즈가 훨씬 더 작을 수도 있을 지라도 전체 프레임 버퍼와 같이 보인다. 디바이스 버퍼 (342) 는 블록-투-라인 (block-to-line) 변환을 용이하게 하고, 블록으로 데이터를 저장하며 라인으로 데이터를 제공한다.

포맷 및 기록 유닛 (344) 은 버퍼 (342) 로부터 픽셀 데이터를 수신하고, 데이터의 수신자 (예를 들어, LCD 유닛 (180) 또는 MDDI 호스트 (146)) 에 의해 기대되는 출력 포맷으로 픽셀 데이터를 포맷하고, LCD 유닛 (180) 으로의 포워딩을 위해 EBI (144) 또는 MDDI 호스트 (146) 에 버스 인터페이스 유닛 (312) 을 통해 포맷된 픽셀 데이터를 제공한다. 유닛 (344) 은 예를 들어, 3 개의 6 비트 기록으로 전송된 18 비트/픽셀, 하나의 16 비트 기록으로 전송된 16 비트/픽셀, MDDI 포맷 등과 같은 다양한 출력 포맷을 지원할 수도 있다. MDDI 호스트 (146) 는 디스플레이 프로세서 (140) 가 그 입력 FIFO (first-in-first-out) 버퍼에 룬을 생성하기 위해 MDDI 호스트를 기다리는 버스 대역폭을 낭비하지 않도록 유닛 (344) 을 스로틀 (throttle) 하는 기능을 가진다.

내부 메모리 (322) 는 디스플레이 프로세서 (140) 내의 다양한 프로세싱 유닛에 의해 이용되는 프로그램 코드 및 데이터를 저장한다. 버스 (328) 는 디스플레이 프로세서 (140) 내의 다양한 유닛을 상호 접속시킨다. 디스플레이 프로세서 (140) 내의 프로세싱 유닛을 이하 더 상세히 설명한다.

1. 프레임 동기화

동기화 유닛 (326) 은 LCD 유닛 (180) 에 데이터의 기록을 제어하여 LCD 스크린상에서 티어링이 발생하지 않도록 보장한다. 현재 프레임에 대한 데이터가 다른 프레임 버퍼로부터 검색되고 LCD 스크린에 제공되는 동안 새로운 프레임에 대한 데이터가 하나의 프레임 버퍼에 기록되도록 2 개의 프레임 버퍼가 자주 이용된다. 2 개의 데이터 프레임은, 데이터가 전체 프레임에 대해 프레임 버퍼로부터 검색되고, 다음 프레임에 대해 다른 프레임 버퍼로부터 검색되며, 그 후, 후속 프레임에 대한 제 1 프레임 버퍼로 되돌아가는 등이 되도록 하는 핑퐁 (ping-pong) 방식으로 작동된다. 새로운 프레임에 대한 데이터는 LCD 스크린에 액세스되지 않는 프레임 버퍼에 기록된다. 2 개의 프레임이 이용되지 않는 경우에 (예를 들어, 메모리 요구사항을 감축시키기 위해), 인위적 영상이 LCD 스크린상에서 나타나지 않기 위해 기록 포인터가 판독 포인터를 패스업 (pass up) 하지 않도록 동기화 유닛 (326) 은 프레임 버퍼 (186) 이 업데이트 되는 것을 보장한다.

도 4a 는 소정의 시간에 LCD 스크린 (188) 을 리프레시 하도록 이용되는 $H \times W$ 프레임 버퍼 (186) 의 콘텐츠를 도시한다. LCD 스크린 (188) 상에서 디스플레이되는 현재 프레임 k 에 대한 픽셀 데이터는 앞선 시기에 프레임 버퍼 (186) 에 기록된다. 리프레시 프로세스는 통상적으로 한번에 한 행씩 프레임 버퍼 (186) 로부터 픽셀 데이터를 판독하고, LCD 스크린 (188) 에 픽셀 데이터를 기록한다. 리프레시 프로세스는 LCD 스크린 (188) 의 최상위로부터 시작하여 스크린의 바닥으로 진행한다. 도 4a 에 도시된 예에 대해, 현재 프레임 k 의 최초 m-1 행은 버퍼 (186) 로부터 판독되고 LCD 스크린 (188) 으로 전송된다. 프레임 k 의 m 번째 행은 LCD 스크린 (188) 상에서 현재 리프레시되고 판독 포인터에 의해 표시된다. 프레임 버퍼 (186) 는 통상적으로 LCD 스크린 (188) 의 리프레시 순서와 동일한 방식으로 구성된다. 그 결과, 판독 포인터가 라인 m 에 있는 경우, 라인 1 내지 m-1 인 프레임 버퍼의 제 1 일부는 이미 이용되고, 아직 LCD 스크린상에 나타나지 않은 새로운 프레임 k+1 에 대한 픽셀 데이터로 업데이트되는 것이 가능하다. 프레임 버퍼 (186) 에서 m 이상의 행은 아직 업데이트되어서는 안되며, 그렇지 않은 경우, LCD 스크린 (188) 은 현재 프레임 k 를 따라 새로운 프레임 k+1 의 일부를 디스플레이 할 것이다.

도 4b 는 그 후의 시간에 프레임 버퍼 (186) 의 콘텐츠를 도시한다. 판독 포인터가 프레임 버퍼 (186) 의 종단에 도달하는 경우에, 이것은 최종 라인으로부터 랩 어라운드 (wrap around) 하고 라인 1 에서 재시작한다. 새로운 프레임 k+1 의 라인

1에 대한 픽셀 데이터는 앞선 시간에서 프레임 버퍼 (186)에 기록된다(도 4a에서). 도 4b에 도시된 예에 대해, 새로운 프레임 k+1의 최초 n-1행은 프레임 버퍼 (186)에 기록되고 LCD 스크린 (188)에 제공될 수 있다. n 이상의 행은 후속 프레임 k+2에 대한 픽셀 데이터로 업데이트 될 수도 있다.

리프레시 사이클은 소정의 프레임의 모든 H 라인의 리프레시이다. 프레임에 대한 최종 라인의 최종 픽셀이 LCD 스크린 (188) 상에서 리프레시된 이후, 판독 포인터는 라인 H로부터 라인 1로 랩 어라운드한다(도 4b에 도시된 바와 같음). 라인 1에 대한 픽셀 데이터는 프레임 버퍼 (186)로부터 다시 검색되고, LCD 스크린 (188)에 기록된다. 판독 포인터가 랩 어라운드하기에 앞서, 프레임 버퍼 (186)의 최상위 행은 새로운 프레임에 대한 픽셀로 업데이트되어야 한다. 그 결과, 프레임 버퍼 (186)는 판독 포인터 위치 이후에(도 4a에서 라인 m 이후 및 도 4b에서 라인 1 이후) 리프레시될 현재 프레임에 대한 픽셀 데이터를 포함해야 한다. 프레임 버퍼 (186)는 판독 포인터 위치 이전에(도 4a에서 라인 m 이전 및 도 4b에서 라인 H 이전) 새로운 프레임에 대한 픽셀 데이터를 포함할 수도 있다.

LCD 스크린 (188)의 리프레시 레이트는 프레임 버퍼 (186)의 업데이트 레이트의 1배 이상(예를 들어, 두배)이 될 수도 있다. 프레임 버퍼 (186)는 LCD 스크린 (188)에 대한 새로운 콘텐츠가 있는 경우에 오직 업데이트된다. 리프레시 레이트가 업데이트 레이트의 복수배인 경우, 프레임 버퍼 (186)는 현재 프레임에 대한 최종 리프레시 사이클동안 업데이트된다. 예를 들어, 리프레시 레이트(예를 들어, 초당 60 Hertz)가 업데이트 레이트(예를 들어, 초당 30 프레임)의 2배인 경우, 업데이트 사이클마다 2개의 리프레시 사이클이 존재한다. 이러한 경우에, 프레임 버퍼 (186)는 새로운 이미지와 함께 티어링을 유발하는, 기록 포인터가 판독 포인터를 패스 업하지 않도록 일 리프레시 사이클보다 긴 레이트로 (1) 업데이트되고, 새로운 프레임에 대한 픽셀 데이터가 지난 프레임과 함께 티어링을 방지하도록 적절히 이용가능하도록 2개의 리프레시 사이클보다 짧은 (2) 레이트로 업데이트된다.

프레임 버퍼 (186)의 업데이트는 판독 포인터에 다소 동기화되어야 한다. 판독 포인터는 보통 LCD 제어기 (182)에 의해 유지되고 LCD 유닛 (180)의 외부 유닛에 통상 이용가능하지 않다. 이러한 경우에, 동기화 유닛 (326)은 판독 포인터의 로컬 복사를 재구성하기 위해 LCD 유닛 (180)으로부터 수직 동기화 신호를 이용할 수도 있다. 수직 동기화 신호는 각각의 리프레시 사이클의 시작에서 펄스를 포함할 수도 있다. 동기화 유닛 (326)은 입력 클럭에 기초하여 증가하는 외부 카운터를 유지할 수도 있고 수직 동기화 신호상의 펄스에 의해 재설정된다. 동기화 유닛 (326)은 프레임 버퍼 (186)에 대한 라인 (H)의 개수의 지식을 가지고 각각의 리프레시 사이클에 대해 근접한 클럭 사이클의 개수를 확인한다. 그 후, 동기화 유닛 (326)은 LCD 스크린 (188)의 하나의 라인을 리프레시 하기 위해 필요한 클럭 사이클의 개수를 추정하여 내부 카운터 값에 기초하여 판독 포인터 위치를 결정할 수 있다. 그 결과, 내부 카운터는 LCD 유닛 (180)의 타이밍을 추정하기 위해 이용될 수 있고 타이밍 오류가 축적되지 않도록 각각의 리프레시 사이클에서 재설정된다. 또한, 동기화 유닛 (326)은 LCD 제어기 (182)로부터 직접 판독 포인터 위치를 획득할 수 있고 그러한 방식으로 판독 포인터를 추적할 수도 있다.

디스플레이 프로세서 (140)는 판독 포인터 위치의 커스하고 일반적인 지식만을 필요로 한다. 예를 들어, 판독 포인터가 라인 118과 121사이의 어딘가에 있다고 알려진 경우, 판독 포인터 위치에 앞서는 몇 개의 라인인 영역(예를 들어, 라인 112에 앞서는 영역)을 업데이트하는 것이 안전하다. 그 결과, 수 퍼센트의 정확성으로의 판독 포인터의 재구성은 LCD 스크린 (188)상의 인위물을 유발하지 않고 프레임 버퍼 (186)를 업데이트하기에 충분할 수도 있다.

2. 뒤집기 및 회전

디스플레이되는 이미지의 자원과 관련하여 상이한 LCD 유닛은 LCD 스크린의 임의의 배향을 가질 수도 있기 때문에, 뒤집기 및 회전은 유용한 기능이다. 예를 들어, 무선 디바이스는 수평 배향을 가지는 LCD 스크린을 가질 수도 있고 애플리케이션은 수직 배향을 기대할 수도 있다. 또 다른 예로서, 셀룰러 폰은 원격전자의 쿨을 위해 이용될 수도 있고 원격 종단(remote end)에 대한 이미지는 폰 제조자와 상이한 제조자로부터의 장비로 인코딩될 수도 있다. 또 다른 예로서, 셀룰러 폰은 너클(knuckle; 플립폰(flip-phone)의 힌지)에 카메라를 가질 수도 있고 LCD 스크린은 폰의 카메라가 LCD 스크린에 관한 거의 모든 임의의 방향으로 포인트 인(point in) 하도록 대략 거의 360°로 비틀릴 수도 있다.

이 실시형태에서, 뒤집기 및 회전 유닛 (330)은 이미지를 4개의 상이한 90°증가 - 0°, 90°, 180° 및 270°로 회전시킬 수 있다. 유닛 (330)은 또한 이미지에 4개의 상이한 뒤집기 옵션 - 뒤집기 없음, 좌-우 뒤집기, 위-아래 뒤집기, 및 좌-우 및 위-아래 모두 뒤집기를 적용할 수도 있다. 그 결과 회전 및 뒤집기의 총 16개의 가능한 조합이 가능하다.

도 5는 16개의 가능한 회전 및 뒤집기 조합에 대한 유일한 이미지 배향을 도시한다. 회전 및 뒤집기의 일부 조합은 동일한 이미지 배향을 발생시키고 그 결과 조합될 수도 있다. 예를 들어, 180°로 회전하고 좌-우 및 위-아래 뒤집기는 무회전

및 뒤집기하지 않은 원래 이미지를 생성한다. 도 5에 도시된 8개의 배향은 모든 16개의 가능한 회전 및 뒤집기 조합을 커버한다. 180° 및 270°로 회전과 4개의 상이한 뒤집기 옵션의 조합은 도 5에 도시된 8개의 이미지 배향에 매핑될 수도 있다.

뒤집기 및 회전 유닛 (330)은 LCD 스크린 (188)으로 포워딩하기에 앞서 이미지의 뒤집기 및 회전을 수행한다. LCD 제어기 (182)는 통상적으로 행으로 픽셀 데이터를 수신하기를 기대한다. 픽셀 데이터 프레임은 또한 보통 메인 메모리 (152)에 행으로 저장된다. 유닛 (330)은 메인 메모리 (152)로부터 픽셀 데이터를 열로 판독함으로써 90° 회전을 수행하고 LCD 제어기 (182)에 픽셀 데이터를 행으로 기록한다. 메인 메모리 (152)는 통상적으로, 메모리 주소의 인접 범위로부터 데이터 정크(chunk)를 폐치하기에 효율적이지만, 약간의 주소 거리로 또 다른 데이터 조각으로부터 이격된 작은 데이터 조각을 폐치하기에 비효율적인 SDRAM이다. 동일한 열에서의 인접한 픽셀이 몇 백 주소 위치로 (또는 행의 폭) 이격될 수도 있기 때문에, 한번에 하나의 열을 메인 메모리 (152)에 액세스하는 것은 상당히 비효율적일 수도 있고 메인 메모리 (152)에 대해 이용가능한 대역폭의 많은 부분을 이용할 수도 있다.

뒤집기 및 회전 유닛 (330)은 픽셀 데이터의 블록 상에서 한번에 동작함으로써 이미지를 효율적으로 뒤집고 회전시킬 수 있다. 유닛 (330)은 프레임 버퍼 (186)의 상부 극좌 (upper leftmost) 코너의 블록으로 시작하는 블록 바이 블록 (block-by-block) 방식으로 메인 메모리 (152)로부터 이미지 데이터를 폐치한다. 유닛 (330)은 메인 메모리 (152)로부터 폐치된 각각의 데이터 블록 상에서 90° 회전을 수행하여 입력 버퍼 (332)에 회전된 데이터 블록을 저장한다. 일단 프레임 버퍼 (186)에 대한 판독 포인트가 회전된 데이터 블록이 위치할 라인을 넘어 진행하는 경우, 디스플레이 프로세서 (140)는 프레임 버퍼에 블록을 기록한다.

도 6은 이미지의 90° 회전을 도시한다. 이 예에 대해, 24개의 블록으로 구성된 이미지는 메인 메모리 (152)에 4×6 블록 배열로 저장된다. 이미지는 90°로 회전되고 프레임 버퍼 (186)에 6×4 블록 배열로 기록된다. 메인 메모리 (152)에서 블록의 각각의 열은 또한 스트립으로 칭한다.

디스플레이 프로세서 (140)는 우선 메인 메모리 (152)로부터 유닛 (330)에 의해 뒤집기 되고 입력 버퍼 (332)에 저장된 블록 1을 폐치한다. 티어링이 발생하지 않도록, 블록 1이 차지할 포인트를 지나쳐 판독 포인트가 진행되는 경우, 디스플레이 프로세서 (140)는 프레임 버퍼 (186)에 회전된 블록 1을 기록한다. 디스플레이 프로세서 (140)는 판독 포인트를 기다릴 필요 없이, 제 1 스트립에 남아있는 블록 2, 3, 및 4 각각을 폐치하고 회전시키며 기록한다. 첫 번째 스트립이 프레임 버퍼 (186)에 기록된 이후에, 디스플레이 프로세서 (140)는 2 번째 스트립의 첫 번째 블록인 블록 5를 폐치하고 회전시킨다 (도 6에 도시된 바와 같음). 다시, 디스플레이 프로세서 (140)는 블록 5가 프레임 버퍼 (186)에서 차지할 포인트를 지나쳐 진행되는 판독 포인트를 기다린다. 판독 포인트가 충분히 멀리 진행한 경우, 디스플레이 프로세서 (140)는 프레임 버퍼 (186)에 블록 5를 기록하고, 그 후, 2 번째 스트립에 남아있는 블록 6, 7, 및 8 각각을 폐치하고 회전시키며 기록한다. 동일한 프로세싱이 이미지에 대해 남아있는 스트립 각각에 대해 수행된다. 디스플레이 프로세서 (140)는 이미지의 최종 스트립에 대한 회전된 블록을 기록하기 이전에 랩 어라운드하는 판독 포인트를 기다린다.

3. 비디오 기능

도 7은 비디오에 보통 이용되는 4:2:0 YCrCb 포맷에 대한 예시적인 8×8 픽셀 블록을 도시한다. YCrCb 포맷에 대해, 컬러 이미지는 (1) 이미지 강도 (또는 블랙/화이트 부분)를 포함하는 휘도 (Y) 성분 및 (2) 이미지에 대한 색 정보를 나르는 레드 (Cr) 및 블루 (Cb) 색도 성분으로 표현된다. 8×8 블록의 64개의 픽셀 각각은 예외한 Y에 의해 나타난 (예를 들어, 8비트) 휘도값에 관련된다. 4:2:0 YCrCb 포맷에 대해, 레드 및 블루 성분은 4개의 픽셀 그룹 각각에 대해 하나의 (예를 들어, 8비트) Cr 값 및 하나의 (예를 들어, 8비트) Cb 값이 존재하도록 서브샘플링된다. Cr 및 Cb 값은 오프-사이트 (off-site) 서브샘플링으로 4개의 픽셀 그룹의 중앙과 (도 7a에 도시됨) 코-사이트 (co-site) 샘플링으로 2개의 수직 픽셀 사이에 (도 7a에 도시되지 않음) 위치한다.

LCD 유닛 (180)은 통상적으로 각각의 픽셀이 RGB 포맷에 존재하고 3개의 색 성분 - 레드 (R), 그린 (G), 및 블루 (B)를 가질것을 기대한다. 각각의 픽셀에 대한 3개의 색 성분은 픽셀에 대한 Y, Cr 및 Cb 값으로부터 획득될 수도 있다. 도 7a에 도시된 4:2:0 YCrCb 포맷에 대해, 디스플레이 프로세서 (140)는 8×8 블록의 4개의 픽셀 그룹 각각에 대해 Cr 및 Cb 값을 업샘플링하여 블록의 8×8 픽셀 위치 각각에서 Cr 및 Cb 값을 획득한다. 각각의 픽셀 위치에서 모든 3개의 Y, Cr 및 Cb 성분을 가지는 이미지는 4:4:4 YCrCb 포맷으로 존재한다.

색도 업샘플링은 다양한 방식으로 수행될 수도 있다. 예를 들어, 4 개의 픽셀 그룹 각각에 대한 Cr 및 Cb 값은 간단히 복사되고 그룹에서 4 개의 픽셀 위치 각각에 대해 Cr 및 Cb 값으로서 이용된다. Cr 값은 또한 필터링될 수도 있고 (1 또는 2 차원으로) 64 개의 픽셀 위치에서 리샘플링된다. 일 실시형태에서, 색 변환 및 스케일링 유닛 (334) 은 쌍선형 필터를 이용하는 픽셀 위치에 가장 인접한 4 개의 Cr 값에 기초하여 8×8 블록의 각각의 픽셀 위치에 대해 Cr 값을 연산한다.

도 7b 는 목적 픽셀 위치에 대해, Cr_x 로 레이블된 (labeled) Cr 값을 획득하기 위해 쌍선형 필터를 이용하는 색도 업샘플링을 도시한다. Cr_x 값을 도출하도록 이용된 4 개의 Cr 값은 목적 픽셀 위치에 4 개의 가장 가까운 Cr 값이고 Cr₁, Cr₂, Cr₃, 및 Cr₄ 로서 레이블된다. Cr_x 값은 다음과 같이 연산된다.

$$Cr_x = \frac{9 \cdot Cr_3 + 3 \cdot Cr_1 + 3 \cdot Cr_4 + Cr_2}{16}$$

식 (1) 에 도시된 바와 같이, Cr₃ 가 폭표 픽셀 위치에 가장 가깝기 때문에, Cr₃ 가 Cr_x 의 연산에서 가장 가중된다. 반대로, Cr₂ 가 폭표 픽셀 위치에 가장 멀기 때문에, Cr₂ 가 Cr_x 의 연산에서 가장 경시된다. 4 개의 가장 가까운 Cr 값에 대한 4 개의 계수 9, 3, 3 및 1 은 Cr_x 가 더 복잡하고 비싼 곱셈기 대신에 간단한 합산 및 시프트 (2 의 제곱승을 곱함에 의함) 로 연산될 수도 있도록 선택된다.

색 변환 및 스케일링 유닛 (334) 은 8×8 블록에서 64 개의 픽셀 위치 각각에 대해 업샘플링된 Cr 값을 획득하는 식 (1) 에 도시된 연산을 수행할 수도 있다. 블록의 4 개의 에지를 따라 픽셀 위치에 대해 업샘플링된 Cr 값을 연산하기 위해, 이웃 블록에 대한 Cr 값은 폐치되고 업샘플링된 Cr 값의 연산에 이용될 수도 있다. 예를 들어, Cr 값의 6×6 블록 (12×12 연장된 이미지 블록에 대한) 은 폐치되고 8×8 블록의 64 개의 픽셀 위치에 대해 업샘플링된 Cr 값을 연산하도록 이용될 수도 있다. 6×6 Cr 블록의 중앙의 16 개의 Cr 값은 현재의 8×8 블록에 속하고, 6×6 블록에 남은 16 개의 Cr 값은 이웃 8×8 블록에 대한 것이다. Cb 값에 대한 업샘플링은 Cr 값에 대한 것과 유사한 방식으로 수행될 수도 있다.

LCD 유닛 (180) 은 통상적으로 RGB 포맷으로 이미지를 받아들이기 때문에, 디스플레이 프로세서 (140) 는 대응하는 RGB 블록을 획득하기 위해 각각 업샘플링된 YCrCb 블록상에서 색 변환을 수행할 수도 있다. YCrCb 로부터 RGB 로의 변환은 다음과 같은 행렬 형태로 나타낼 수도 있다.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.1644 & 0 & 1.596 \\ 1.1644 & -0.3918 & -0.813 \\ 1.1644 & 2.0172 & 0 \end{bmatrix} \cdot \left(\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \right)$$

식 (2) 는 공개적으로 입수 가능한 ITU-R 발행물 601 (권고 601) 에 개시된다.

색 변환 및 스케일링 유닛 (334) 은 각각의 픽셀 위치에 대한 Y, Cr, 및 Cb 값에 색 변환을 수행할 수도 있어 그 픽셀 위치에 대응하는 R, G, 및 B 값을 획득한다. 식 (2) 에 도시된 연산은 계수 (예를 들어, 12 비트), 입력 YCrCb 성분 (예를 들어, 8 비트), 및 출력 RGB 성분 (예를 들어, 6 또는 8 비트) 에 대한 적절한 솔루션을 가지는 곱셈기로 수행될 수도 있다.

디스플레이 프로세서 (140) 는 또한 이미지를 사이즈상 위 아래로 스케일링할 수도 있다. 스케일링 (또는 리샘플링) 은 예를 들어, 줌인 또는 줌아웃 (예를 들어, 카메라 뷰파인더 (viewfinder)), LCD 스크린상에 복수의 이미지 위치시키기 (예를 들어, 섬네일 크기의 사진 또는 원격전화회의 콜) 등과 같은 다양한 목적을 위해 이용될 수도 있다. 스케일링은 4:2:0 YCrCb 포맷, 4:4:4 YCrCb 포맷, 또는 다른 포맷인 이미지상에서 수행될 수도 있다. 일반적으로, 이미지는 임의의 M/L 비율로 스케일링될 수도 있고, 여기서 M 은 새로운 솔루션의 수평 픽셀의 개수이고 L 은 지난 솔루션의 수평 픽셀의 개수이다. 그러나, 연산은 2 의 제곱승 예를 들어, 2, 4, 8, 16 등에 의해 간단화되고 스케일링된다.

L 이 2, 4, 8, 16 등이 될 수도 있는 1/L 다운스케일링을 위해 소정의 성분 X 에 대한 연산은,

$$X_{avg} = \frac{1}{L \cdot L} \cdot \sum_{j=1}^L \sum_{i=1}^L X(i, j)$$

과 같이 나타낼 수도 있고, 여기서 X 는 4:4:4 RGB 포맷에 대한 레드, 그린 또는 블루 성분 또는 4:2:0 YCrCb 포맷에 대한 휘도를 나타내고;

$X(i, j)$ 는 픽셀 위치 (i, j) 에서의 성분값이며;

X_{avg} 는 $L \times L$ 박스에 대한 평균 성분값이다.

Cr 및 Cb 값의 하나의 세트는 4:2:0 YCrCb 포맷에 대한 4 개의 픽셀 그룹 각각에 대해 이용가능하기 때문에, $1/L$ 로 다운스케일링을 위한 색도 성분에 대한 연산은 L 값에 의존한다.

도 8a 는 8×8 블록에 대한 $1/2$ 다운스케일링을 도시한다. 각각의 2×2 박스의 4 개의 휘도값은 합산되고 4 로 나누어져 (또는 2 비트 우측으로 시프트됨) 박스에 대한 평균 Y 값을 획득한다. Cr 및 Cb 값의 하나의 세트는 이미 4:2:0 YCrCb 포맷에 대한 4 개의 픽셀 그룹 각각에 이미 제공되었기 때문에 Cr 및 Cb 성분에 대해 어떠한 연산도 필요하지 않다.

$1/4$ 다운스케일링을 위해, 각각의 4×4 박스에서 16 개의 휘도값이 합산되고 16 으로 나누어져 박스에 대한 평균 Y 값을 획득한다. 각각의 4×4 박스에 대한 4 개의 Cr 값은 합산되고 4 로 나누어져 박스에 대한 평균 Cr 값을 획득한다. 각각의 4×4 박스에 대한 4 개의 Cb 값은 또한 합산되고 4 로 나누어져 박스에 대한 평균 Cb 값을 획득한다. 색도 업샘플링은 $1/4$ 이상으로 다운스케일링을 수행하는 경우에 필요하지 않다.

$1/8$ 다운스케일링을 위해, 8×8 블록에서 64 개의 휘도값이 합산되고 64 로 나누어져 블록에 대한 평균 휘도값을 획득한다. 8×8 블록에 대한 16 개의 Cr 값은 합산되고 16 으로 나누어져 평균 Cr 값을 획득한다. 8×8 블록에 대한 16 개의 Cb 값은 또한 합산되고 16 으로 나누어져 평균 Cb 값을 획득한다. 색도 업샘플링은 $1/8$ 다운스케일링에 필요하지 않다.

도 8b 는 8×8 블록에 대한 $3/8$ 다운스케일링을 도시한다. 8×8 블록의 64 개의 픽셀은 예를 들어, 도 8b 에 도시된 바와 같이 9 개의 박스로 배열된다. 각각의 박스는 9, 6, 또는 4 개의 픽셀을 포함한다. 각각의 박스에서의 휘도값은 합산되고 박스에서의 픽셀 개수로 나누어져 박스에 대한 평균 Y 값을 획득한다. 색도 성분은 각각의 픽셀에 대한 Cr 값 및 Cb 값을 획득하기 위해 업샘플링될 수도 있다. 각각의 색 성분에 대한 색도값은 휘도 성분에 대한 것과 동일한 방식으로 합산되고 나누어진다.

도 8c 는 8×8 블록에 대한 $5/8$ 다운스케일링을 도시한다. 8×8 블록의 64 개의 픽셀은 예를 들어, 도 8c 에 도시된 바와 같이 25 개의 박스로 배열된다. 각각의 박스는 4, 2, 또는 1 개의 픽셀을 포함한다. 각각의 박스에서의 휘도값은 합산되고 박스에서의 픽셀 개수로 나누어져 박스에 대한 평균 Y 값을 획득한다. 다시, 각각의 색도 성분은 우선 업샘플링되고 휘도 성분에 대한 것과 동일한 방식으로 합산되고 나누어질 수도 있다.

일반적으로, M/L 다운스케일링은 $N \times N$ 블록에 대해 $N \times N$ 블록내의 $N \cdot N$ 픽셀을 적절한 박스 개수로 배열하여 다운스케일링을 달성함으로써 수행될 수도 있다. 상이한 사이즈의 박스는 다운스케일링된 이미지에서 인위물을 감소시키기 위해 $N \times N$ 블록의 가능한 범위까지 균등하게 분배될 수도 있다. 이웃 $N \times N$ 블록에 대한 색도값은 다운스케일링된 이미지에 대한 평균 Cr 및 Cb 값을 연산하기 위해 폐지되고 이용될 수도 있다.

2 배로 업스케일링 (또는 업샘플링) 하기 위해, 수직과 수평방향 모두에 존재하는 각각의 픽셀쌍 사이에 새로운 픽셀이 삽입된다. $N \cdot N$ 픽셀을 가지는 $N \times N$ 블록은 $4 \cdot N \cdot N$ 픽셀, 또는 원래 블록의 4 배의 픽셀 개수를 가지는 $2N \times 2N$ 블록을 획득하기 위해 2 배로 업스케일링될 수도 있다. 삽입은 다양한 방식으로 수행될 수도 있다. 일 실시형태에서, 바이큐빅 삽입은 휘도를 위해 이용되고, 쌍선형 삽입은 Cr 및 Cb 성분을 위해 이용된다. $N \times N$ 블록의 에지에 위치하는 픽셀을 포함하는 $N \times N$ 블록의 모든 픽셀에 바이큐빅 및 쌍선형 내삽을 수행하기 위해, $(N+2) \times (N+2)$ 블록은 메인 메모리 (152) 로부터 폐지될 수도 있다.

도 9a 는 소정의 행에서 X 값에 대해 바이큐빅 삽입을 도시한다. 행 i 에 대해 삽입된 X 값은 다음과 같이 나타난다:

$$X_{in}(i, j) = \frac{-X(i-3, j) + 9X(i-1, j) + 9X(i+1, j) - X(i+3, j)}{16}$$

여기서 $X(i, j-x)$ 는 $X(i, j)$ 의 왼쪽으로 x 하프 (half) 픽셀 위치에 위치한 X 값이고;

$X(i, j+x)$ 는 $X(i, j)$ 의 오른쪽으로 x 하프 픽셀 위치에 위치한 X 값이다.

식 (4) 및 도 9a 에 도시된 바와 같이, 행에 각각 삽입된 X 값은 동일한 행에 삽입된 X 값의 오른쪽으로 2 개의 X 값과 왼쪽으로 2 개의 X 값에 의해 결정된다.

도 9b 는 소정의 열에서 X 값에 대해 바이큐빅 삽입을 도시한다. 열 j 에 대해 삽입된 X 값은 다음과 같이 나타낸다:

$$X_{\text{int}}(i, j) = \frac{-X(i-3, j) + 9X(i-1, j) + 9X(i+1, j) - X(i+3, j)}{16}$$

여기서, $X_{\text{int}}(i, j)$ 는 픽셀 위치 (i, j) 에 삽입된 X 값이고;

$X(i-x, j)$ 는 $X(i, j)$ 위로 x 하프-픽셀 위치에 위치한 X 값이며;

$X(i+x, j)$ 는 $X(i, j)$ 아래로 x 하프-픽셀 위치에 위치한 X 값이다.

식 (5) 및 도 9b 에 도시된 바와 같이, 행에 각각 삽입된 X 값은 동일한 열에 삽입된 X 값의 위의 2 개의 X 값과 아래의 2 개의 X 값에 의해 결정된다.

$N \times N$ 블록의 R , G , B , 및 Y 에 대한 업스케일링은 수평방향으로 삽입된 픽셀값을 획득하기 위해 블록의 각각의 행을 삽입하고 (1), 수직 방향으로 삽입된 픽셀값을 획득하기 위해 삽입된 행의 각각의 열을 삽입함으로써 (2) 수행될 수도 있다. Cr 및 Cb 에 대한 업스케일링은 각각의 원래 픽셀 위치에 대한 Cr 값 및 Cb 값을 획득하기 위해 색도를 업샘플링하고 (1), 각각의 원래 픽셀 위치에서의 Cr 및 Cb 값을 원래 픽셀 위치의 오른쪽으로 하프 픽셀, 아래로 하프 픽셀, 오른쪽으로 하프 픽셀 및 아래로 하프 픽셀에 존재하는 3 개의 삽입된 픽셀 위치에 복사함으로써 (2) 수행될 수도 있다. 업스케일링은 또한 다른 방식으로 수행될 수도 있다. 업샘플링의 결과는 업샘플링 블록의 각각의 픽셀 위치에 대한 R , G , 및 B 값 세트 또는 Y , Cr 및 Cb 값 세트이다.

4. 그래픽 기능

디스플레이 프로세서 (140) 는 필요에 따라, 픽셀 데이터 블록을 크롭할 수도 있다. 전체 블록은 효율적인 메모리 액세스를 위해 검색될 수도 있다. 그러나, 블록의 일부만이 프로세싱되고 프레임 버퍼 (186) 에 제공된다. 이러한 경우에, 믹스, 블렌드, 및 크롭 유닛 (340) 은 프로세싱될 필요없는 부분을 클립하거나 잘라내고 원하는 부분을 보유한다.

디스플레이 프로세서 (140) 는 단일의 출력 이미지를 생성하기 위해 2 개의 이미지를 조합하는 알파-블렌딩을 수행할 수도 있다. 알파-블렌딩은 2 개의 이미지를 서로의 상부상에 중첩하는데, 예를 들어, 비디오 또는 그래픽 위에 텍스트를 위치시키거나, 비디오 위에 그래픽을 나타내는 등을 하는데 이용될 수도 있다. 알파-블렌딩은 다음과 같은 각각의 성분 X 에 대해 수행될 수도 있다.

$$X_{\text{out}}(i, j) = \alpha \cdot X_1(i, j) + (1 - \alpha) \cdot X_2(i, j)$$

여기서, $X_1(i, j)$ 는 블록의 픽셀 위치 (i, j) 에 대한 제 1 버퍼 (336) 로부터의 값이고;

$X_2(i, j)$ 는 픽셀 위치 (i, j) 에 대한 제 2 버퍼 (338) 로부터의 X 값이며;

α 는 $X_1(i, j)$ 및 $X_2(i, j)$ 에 대한 가중치를 결정하는 계수이고;

$X_{\text{out}}(i, j)$ 는 픽셀 위치 (i, j) 에 대한 출력 Y 값이다.

통상적으로, $0 \leq \alpha \leq 1$ 이다. 더 큰 α 값은 $X_1(i, j)$ 에 더 많은 가중치를 주고, 더 작은 α 값은 $X_2(i, j)$ 에 더 많은 가중치를 준다.

믹스, 블렌드, 및 크롭 유닛 (340) 은 식 (6) 에서 도시된 바와 같이 출력 이미지에서 각각의 픽셀 위치에 대해 알파-블렌딩을 수행할 수도 있다. 유닛 (340) 은 동일한 픽셀 위치에 대해 버퍼 (336 및 338) 에서 2 개의 X 값을 스케일링하고 결합하며 디바이스 버퍼 (342) 에 결과 출력 X 값을 저장한다. 버퍼 (336 및 338) 는 알파-블렌딩을 위한 전면 (foreground) 과 후면 (background) 각각을 위해 2 개의 블록을 저장할 수도 있다. 또한, 단일의 버퍼는 알파-블렌딩을 위해 이용될 수도 있다. 전면 블록이 검색되는 경우, 스케일링된 전면 블록, $\alpha \cdot X_1(i, j)$ 는 제 1 버퍼 (336) 에 저장될 수도 있다. 후면 블록이 검색되는 경우, 스케일링된 후면 블록 $(1-\alpha) \cdot X_2(i, j)$ 은 저장되고 스케일링된 전면 블록과 결합될 수도 있다.

디스플레이 프로세서 (140) 는 후면 이미지만, 전면 이미지만, 또는 후면 및 전면 이미지 모두의 결합을 나타내는 투명성을 수행할 수도 있다. 투명성은 예를 들어, 후면 이미지 위의 아이콘 또는 텍스트를 나타내기 위해 이용될 수도 있다. 믹스, 블렌드, 및 크롭 유닛 (340) 은 출력 이미지에서의 각각의 픽셀 위치에 대해 투명성을 수행할 수도 있다. 유닛 (340) 은 소정의 픽셀 위치에 대한 (예를 들어, 후면 이미지에 대한) 픽셀값 및 동일한 픽셀 위치에 대한 (예를 들어, 전면 이미지에 대한) 또 다른 픽셀값을 검색한다. 제 2 버퍼 (338) 로부터 검색된 픽셀값이 투명성값과 매치하는 경우, 유닛 (340) 은 제 1 버퍼 (336) 로부터의 픽셀값을 디바이스 버퍼 (342) 에 기록한다. 그렇지 않은 경우, 유닛 (340) 은 제 2 버퍼 (338) 로부터의 픽셀값을 디바이스 버퍼 (342) 에 기록한다. 버퍼 (338) 내의 픽셀값은 B 비트 (예를 들어, $B = 18$) 로 구성되고 0 내지 $2^B - 1$ 의 범위내에서 감소한다. 2^B 의 가능한 값 중 하나는 픽셀값을 디스플레이할지 여부를 나타내기 위해 보유하고 투명성값으로서 이용될 수도 있다. 그 결과, 투명성 값은 마스크 또는 "열쇠 (key)" 로서 이용된다. 동일하거나 상이한 투명성 값이 상이한 블록에 대해 이용될 수도 있다. 유닛 (340) 에 각각의 블록에 적용가능한 투명성 값이 알려진다.

디스플레이 프로세서 (140) 는 또한 알파-블렌딩과 함께 투명성을 수행할 수도 있다. 각각의 픽셀 위치에 대해, 제 2 버퍼 (338) 로부터의 픽셀값이 투명성 값과 매치하는 경우, 제 1 버퍼 (336) 로부터의 픽셀값이 디바이스 버퍼 (342) 에 제공된다. 제 2 버퍼 (338) 로부터의 픽셀값이 투명성 값과 매치하는 않는 경우, 버퍼 (336 및 338) 로부터의 2 개의 픽셀값의 알파-블렌딩이 디바이스 버퍼 (342) 에 제공된다.

텍스트 오버레이는 투명성의 특정한 경우이다. 텍스트 오버레이에서, 제 2 버퍼의 픽셀값은 세트 색에 대한 값 또는 투명성 값 중 하나이다. 제 2 버퍼 (338) 로부터의 픽셀값이 세트 색과 일치하는 경우, 디바이스 버퍼 (342) 에 제공되고 그렇지 않은 경우, 제 1 버퍼 (336) 로부터 대응하는 픽셀값이 디바이스 버퍼 (342) 에 제공된다.

디스플레이 프로세서 (140) 는 픽셀당 하나의 비트 (bpp) 값을 픽셀당 복수의 비트값으로 변환하기 위하여 폰트 (font) 확장을 수행할 수도 있다. 하나의 bpp 값은 텍스트를 효율적으로 나타내기 위하여 이용될 수도 있다. 복수의 bpp 값은 예를 들어, 레드에 대한 6 비트, 그린에 대한 6 비트, 및 블루에 대한 6 비트를 포함하는 18 비트 값일 수도 있다. 복수의 bpp 값은 텍스트에 대한 폰트 색 세트 또는 투명성 값 중 하나이다. 복수의 bpp 값은 제 2 버퍼 (338) 에 기록될 수도 있다.

폰트 확장 및 투명성의 결합은 디스플레이 프로세서 (140) 가 제 1 버퍼 (336) 으로부터의 각각의 1-bpp 값을 검사하고, 그 후, 제 2 버퍼 (338) 로부터 대응하는 픽셀값 또는 선택된 폰트 색 중 하나를 디바이스 버퍼 (342) 에 기록하도록 한다.

명료함을 위해, 특정한 기능 세트를 수행하기 위해 설계된 디스플레이 프로세서 (140) 의 특정 실시형태가 상술되었다. 또한, 디스플레이 프로세서 (140) 의 특정 설계는 설명되었고 도 3 에 도시되었다. 일반적으로, 디스플레이 프로세서는 프레임 버퍼로 이미지 데이터의 전송을 용이하게 할 수 있는 임의의 기능 및 임의의 개수의 기능들을 수행할 수도 있다. 그 결과, 디스플레이 프로세서는 도 3 에 도시된 것들보다 더 적은, 더 많은, 및/또는 상이한 프로세싱 유닛으로 설계될 수도 있다.

디스플레이 프로세서 (140) 는 다양한 이점을 제공할 수 있다. 통상적인 디스플레이 프로세서 (140) 는 LCD 스크린상에 비디오, 그래픽, 텍스트 등을 제공하는 복수의 프로세서들을 효율적으로 서빙한다. 이것은 리턴던트 회로 구성요소 양을 감소시킨다. 디스플레이 프로세서 (140) 는 태스크를 포스트-프로세싱하는 것으로부터 이들 프로세서들을 경감시킬 수 있고, 이는 이들 프로세서들이 그들 자신의 특정 태스크에 집중할 수 있게 한다. 디스플레이 프로세서 (140) 는 이들 프로세서들에 대한 LCD 유닛 (180) 에 이미지 데이터를 기록할 수 있다. LCD 유닛은 보통 느리기 때문에, 이들 프로세서는 LCD 유닛이 이 이미지 데이터를 받아들이기를 기다리며 다수의 클럭 사이클을 낭비할 수도 있다. 디스플레이 프로세서 (140) 는 상술한 바와 같이, LCD 유닛에 이미지 데이터를 효율적으로 기록할 수 있다.

디스플레이 프로세서 (140) 는 프레임 버퍼 (186) 에 이미지를 전송하기에 앞서 이미지상에 대해 부산하게 포스트-프로세싱 (예를 들어, 되집기, 회전, 색 변환, 리스케일링 등) 을 수행할 수 있다. 이것은, 프로세서가 순차적으로 폐지되고 LCD

스크린상에서 디스플레이 되지 않는 이미지에 대해 포스트-프로세싱을 수행하는 상황을 방지한다. 예를 들어, 프레임은 MPEG 비디오에 대해 파손되어 검색될 수도 있고, 포스트-프로세싱은 다양한 원인, 예를 들어, 비디오가 오디오를 지연시키기 때문에 디스플레이되지 않는 프레임에 대해 스킵될 수도 있다.

디스플레이 프로세서 (140) 는 효율적인 방식으로 포스트-프로세싱 태스크를 수행할 수 있다. 예를 들어, 다른 프로세서들은 포스트-프로세싱 태스크의 일부 (예를 들어, 오버레이) 를 수행하기 위해 통상 메인 메모리 (152) 의 복수의 액세스를 요구할 수도 있다. 반대로, 디스플레이 프로세서 (140) 는 더 적은 (예를 들어, 하나) 메모리 액세스로 이들 태스크를 수행할 수도 있다. 디스플레이 프로세서 (140) 는 그 네이티브 포맷내의 이미지 데이터를 판독하고 디스플레이될 프레임을 프로세싱할 수 있다. 디스플레이 프로세서 (140) 는 또한 데이터 블록상에서 동작하고, 이는 메모리 액세스 및 90°회전과 같은 동작에 대해 훨씬 더 효율적일 수도 있다. 디스플레이 디바이스 (140) 는 LCD 유닛 (180) 에 의해 기대되는 출력 포맷 및 라인으로 출력 이미지 데이터를 제공한다. 디스플레이 프로세서 (140) 는 프레임 버퍼로의 기록에 관련된 오버헤드를 감소시키기 위해 이미지 데이터의 전체 프레임이 프레임 버퍼 (186) 에 기록되고 있다는 것을 LCD 제어기 (182) 에 알려준다. 디스플레이 프로세서 (140) 는 입력 이미지 데이터를 검색하고, 포스트-프로세싱을 부산하게 수행하여 LCD 제어기에 의해 기대되는 포맷으로 티어링을 유발하지 않고 프레임 버퍼 (186) 에 출력 이미지 데이터를 제공한다.

여기서 설명된 디스플레이 프로세서는 다양한 방식으로 구현될 수도 있다. 예를 들어, 디스플레이 프로세서는 하나 이상의 주문형 집적회로 (ASIC), 디지털 신호 프로세서 (DSP), 디지털 신호 프로세싱 디바이스 (DSPD), 프로그래머블 로직 디바이스 (PLD), 필드 프로그래머블 게이트 어레이 (FPGA), 프로세서, 제어기, 마이크로-제어기, 마이크로프로세서, 여기에 설명된 기능을 수행하도록 설계된 다른 전자 유닛, 또는 그의 조합내에서 구현될 수도 있다.

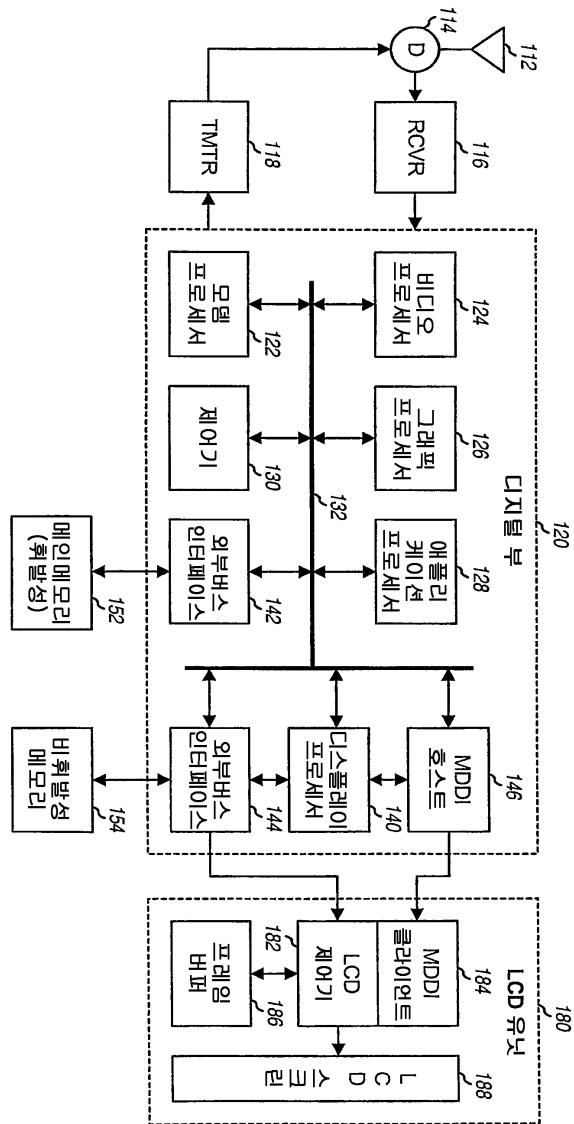
디스플레이 프로세서에 의해 지원되는 기능들은 하드웨어, 소프트웨어, 두 개의 조합으로 또한 구현될 수도 있다. 예를 들어, 도 3 에 도시된 다양한 프로세싱 유닛은 전용 하드웨어로 구현될 수도 있다. 또한, 하나 이상의 이들 프로세싱 유닛들은 여기서 설명된 기능들을 수행하는 소프트웨어 모듈 (예를 들어, 절차, 기능 등) 로 구현될 수도 있다. 소프트웨어 코드는 메모리 유닛 (예를 들어, 도 3 의 메모리 유닛 (322)) 에 저장되고 프로세서 (예를 들어, 명령 프로세서 (320)) 에 의해 실행될 수도 있다.

표제 (heading) 는 참조를 위해 및 일정한 섹션의 위치시키기를 돕기 위해 여기에 포함된다. 이들 헤딩들은 여기에서 설명된 개념의 범위를 제한하는 것으로 의도되지는 않으며, 이들 개념들은 전체 명세서 전반에 걸친 다른 섹션에서 이용가능성을 가질 수도 있다.

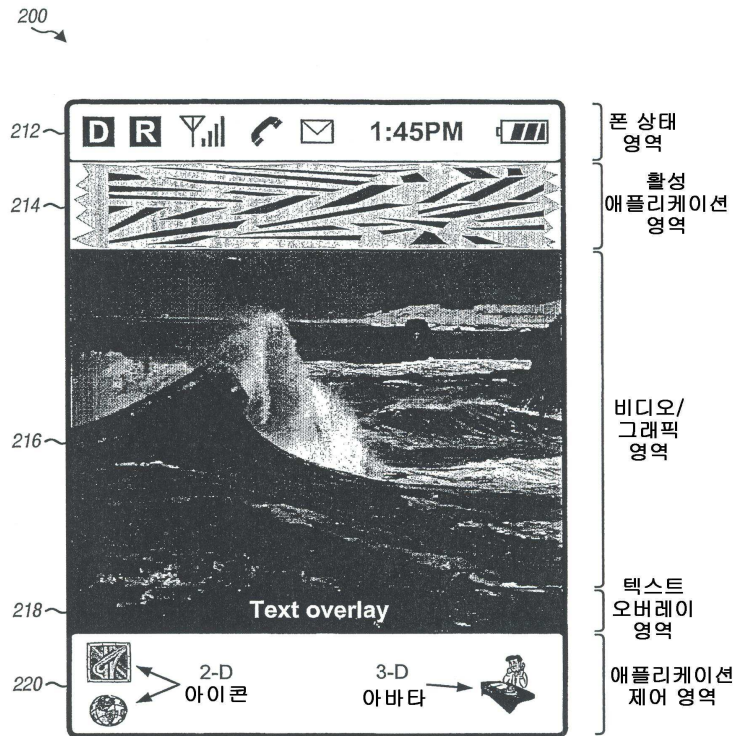
개시된 실시형태의 이전의 설명은 당업자가 본 발명을 수행 또는 사용할 수 있도록 제공된다. 이들 실시형태에 대한 다양한 변형은 당업자에게는 용이하게 명백할 것이며, 여기에 정의된 일반적인 원리는 본 발명의 범위 또는 사상을 벗어나지 않고 다른 실시형태에 적용될 수도 있다. 따라서, 본 발명은 여기에서 설명된 실시형태로 제한하려는 것이 아니라, 여기에 개시된 원리 및 신규한 특성에 부합되는 최광의 범위를 부여하려는 것이다.

도면

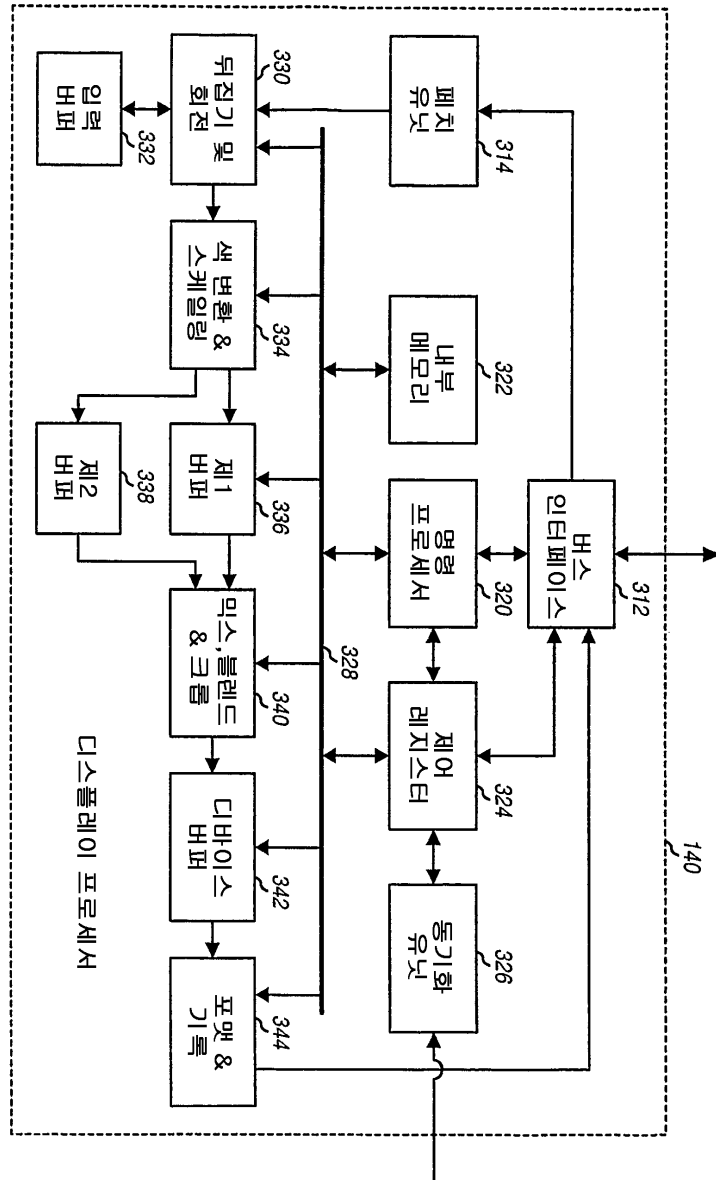
도면1



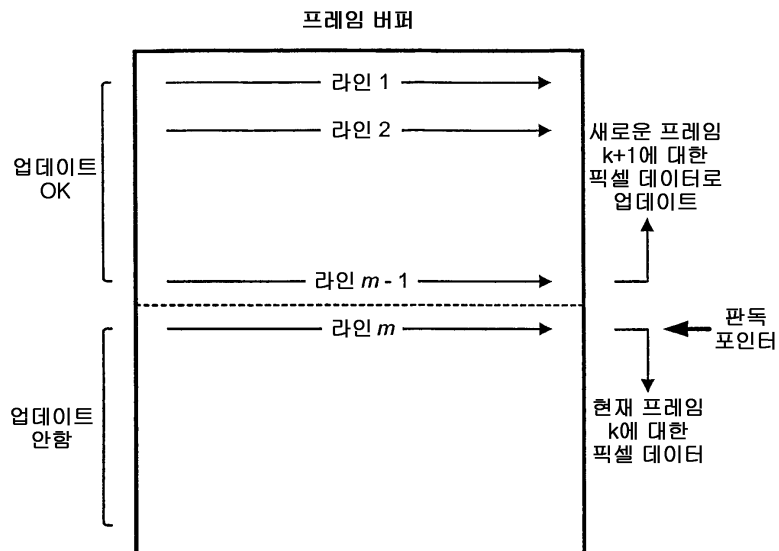
도면2



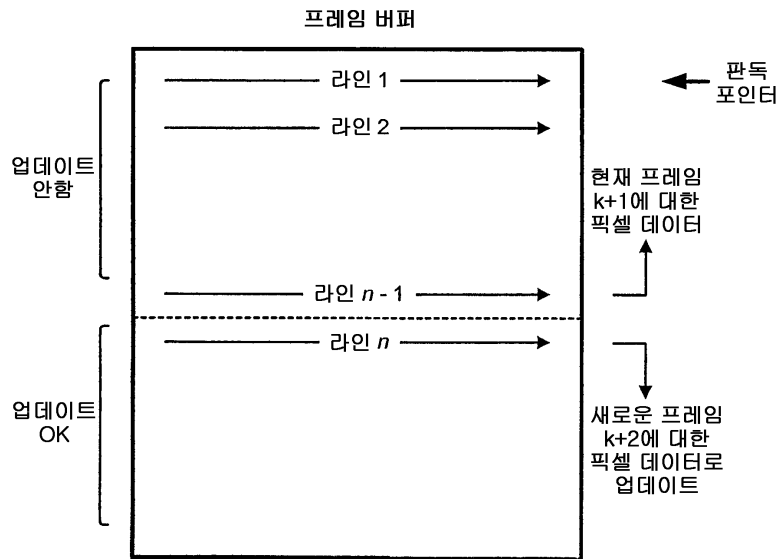
도면3



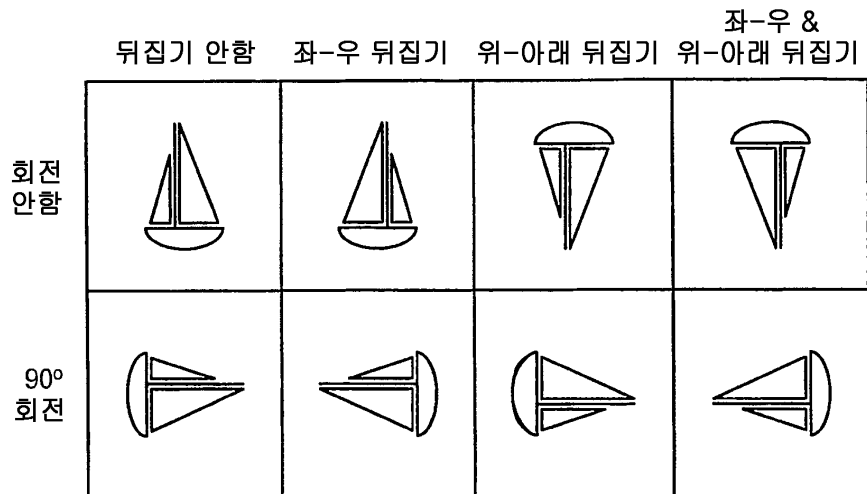
도면4a



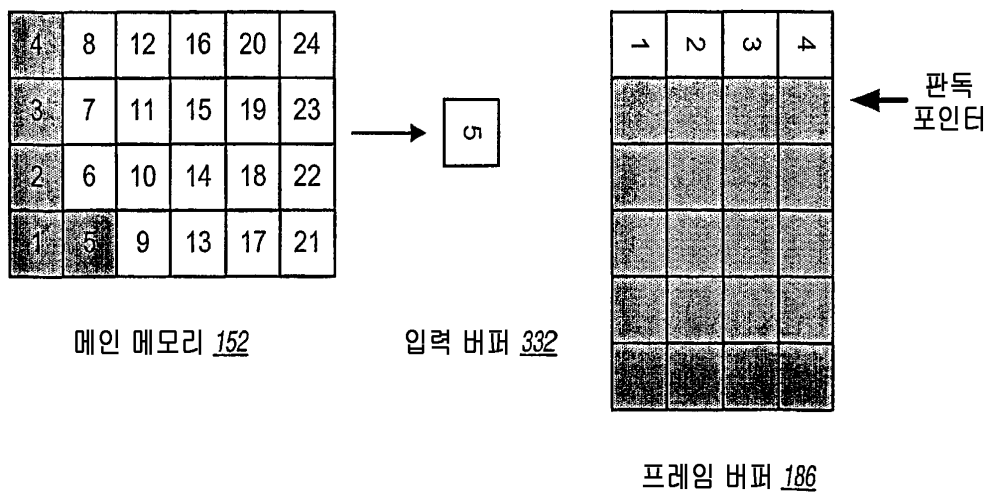
도면4b



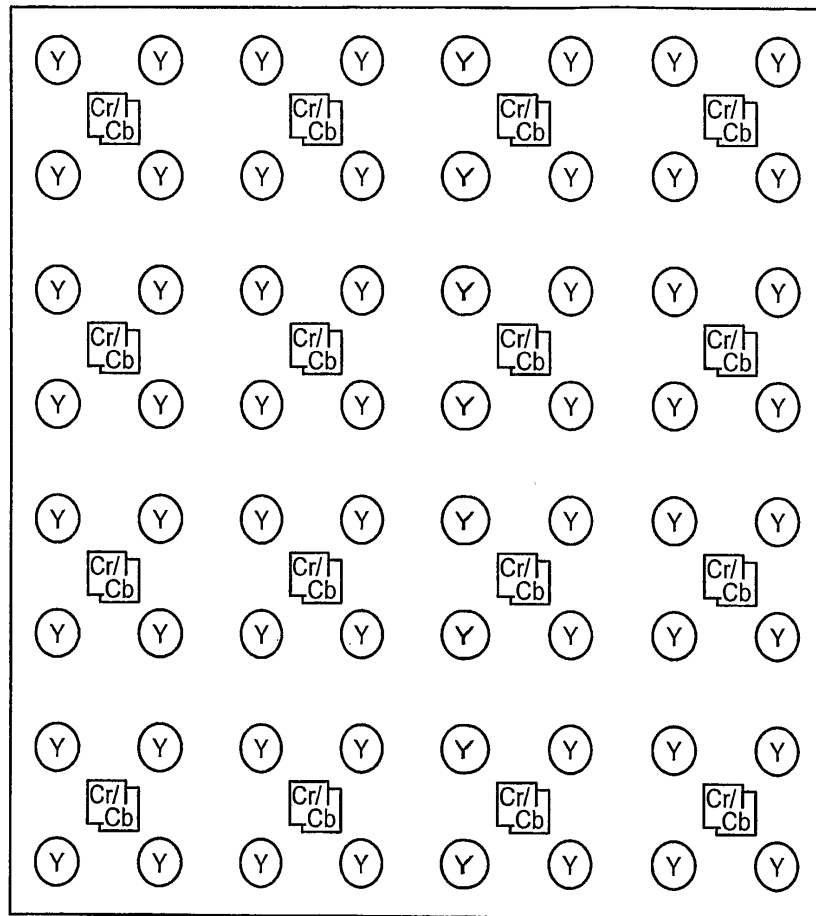
도면5



도면6

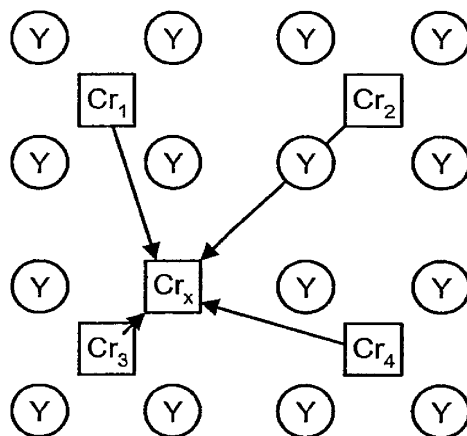


도면7a



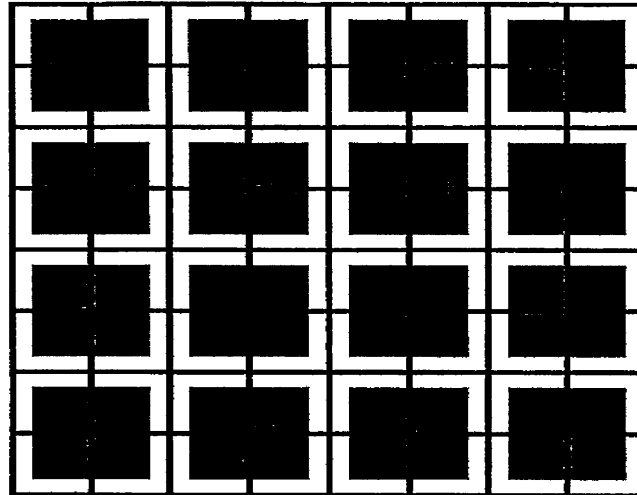
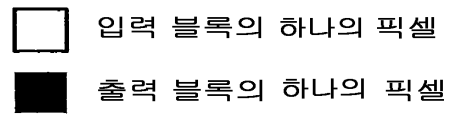
도면7b

색도 업샘플링



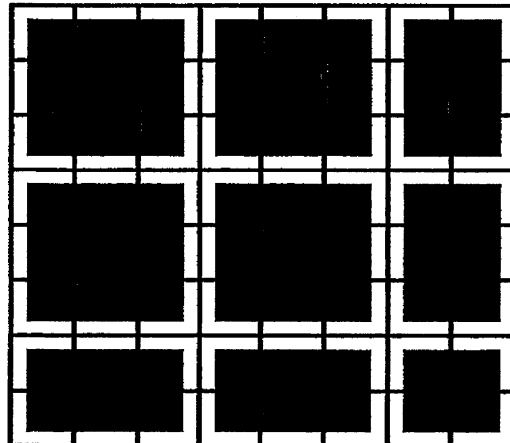
도면8a

1/2 다운스케일링



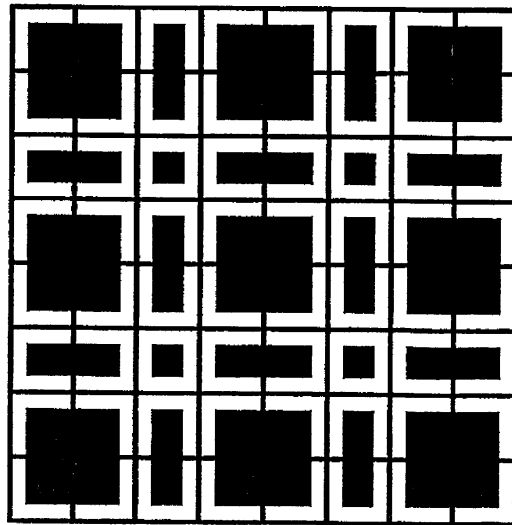
도면8b

3/8 다운스케일링

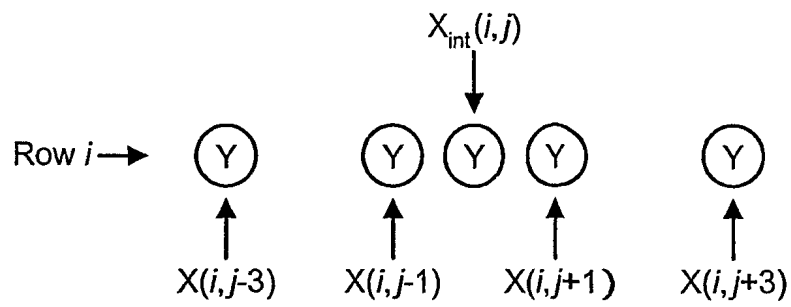


도면8c

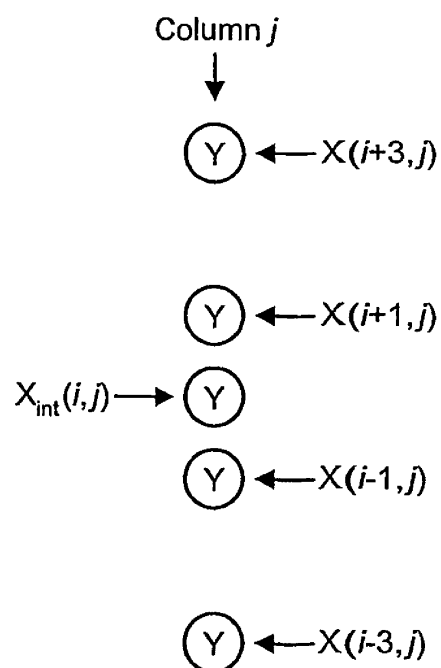
5/8 다운스케일링



도면9a



도면9b



专利名称(译)	显示无线设备的处理器		
公开(公告)号	KR1020070022236A	公开(公告)日	2007-02-26
申请号	KR1020067019717	申请日	2005-02-24
[标]申请(专利权)人(译)	高通股份有限公司		
申请(专利权)人(译)	美国高通公司，		
当前申请(专利权)人(译)	美国高通公司，		
[标]发明人	LUDWIN ALBERT SCOTT 루드윈앨버트스콧 KING SCOTT HOWARD 킹스콧하워드		
发明人	루드윈앨버트스콧 킹스콧하워드		
IPC分类号	G09G3/36 G09G5/393 G09G3/20 G06F15/16 G09G5/00 G09G5/02 G09G5/36 G09G5/39 G09G5/397 H04M1/73		
CPC分类号	G09G3/3611 G09G5/006 G09G5/02 G09G5/363 G09G5/39 G09G5/393 G09G5/397 G09G2340/0407 G09G2340/0492 G09G2340/10 G09G2340/125 H04M1/72544 H04W52/027 Y02D70/122		
优先权	10/954408 2004-09-29 US 60/547711 2004-02-24 US 60/569288 2004-05-06 US		
其他公开文献	KR100910135B1		
外部链接	Espacenet		

摘要(译)

显示处理器包括接口单元，指令处理器，同步单元，至少一个处理单元和设备缓冲器。接口单元接收输入图像数据（例如，从主存储器）并提供帧缓冲器的输出图像数据。指令处理器接收指令（例如，在脚本或列表中）并指示处理单元的操作。同步单元确定帧缓冲器的读指针的位置，并控制输出图像数据到帧缓冲器的写入，以避免在LCD屏幕上引起视觉伪像。处理单元可以执行各种后处理功能，例如区域翻转，区域旋转，两种视频格式之间的颜色转换（例如，从YCrCb到RGB），上/下图像尺寸重新缩放， α 混合，透明度，文本叠加等。

