(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년07월27일
(11) 등록번호 10-2426012
(24) 등록일자 2022년07월22일

(51) 국제특허분류(Int. Cl.)
G06F 9/451 (2018.01) *G06F 9/4401* (2018.01)
G09G 5/14 (2006.01) *G09G 5/36* (2006.01)
G09G 5/399 (2006.01)
(52) CPC특허분류
G06F 9/451 (2018.02)
G06F 9/4411 (2013.01)
(21) 출원번호 10-2019-7006548
(22) 출원일자(국제) 2017년10월11일
심사청구일자 2020년08월06일
(85) 번역문제출일자 2019년03월05일
(65) 공개번호 10-2019-0073350
(43) 공개일자 2019년06월26일
(86) 국제출원번호 PCT/CA2017/051206
(87) 국제공개번호 WO 2018/076102
국제공개일자 2018년05월03일
(30) 우선권주장
15/338,492 2016년10월31일 미국(US)
(56) 선행기술조사문헌
KR1020160019896 A*
(뒷면에 계속)
전체 청구항 수 : 총 24 항 심사관 : 최정권

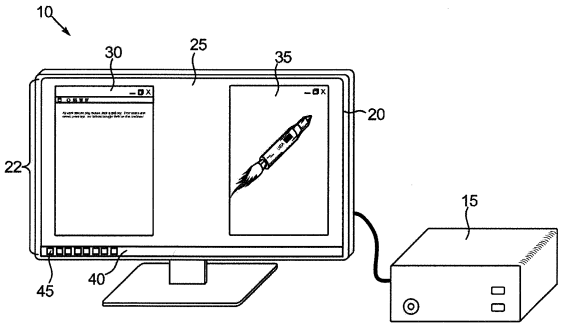
(73) 특허권자
에이티아이 테크놀로지스 유엘씨
캐나다 온타리오 엘3티 7엑스6 마크햄 커머스 밸리 드라이브 이스트 1
(72) 발명자
쿠 앤서니 울
캐나다 온타리오 엘3티7엑스6 마크햄 원 커머스 밸리 드라이브 이스트
후세인 사이에드 아타르
캐나다 온타리오 엘3티7엑스6 마크햄 원 커머스 밸리 드라이브 이스트
(74) 대리인
박장원

(54) 발명의 명칭 **데스크탑 환경에서 애플리케이션 렌더링에서 온 스크린까지 시간을 동적으로 감소시키기 위한 방법 장치**

(57) 요 약

디스플레이 디바이스(20) 상에 디스플레이하기 위해 비디오 컨텐츠를 렌더링하고 주기적으로 상기 디스플레이 디바이스를 리프레시하도록 동작가능한 컴퓨팅 디바이스(15)를 포함하는 시스템이 제공된다. 비디오 컨텐츠는 적어도 하나의 애플리케이션 윈도우(35)를 포함한다. 데스크탑 합성자(90)는 상기 적어도 하나의 애플리케이션 윈도우를 포함하는 합성된 외관들인 비디오 프레임들을 합성하고 상기 비디오 프레임들을 상기 디스플레이 디바이스로 전달하기 위해 버퍼 플립을 개시하는 명령들을 웨이크하고 실행하도록 동작가능하다. 고해상도 타이머(92)는 상기 데스크탑 합성자가 디스플레이 리프레시들 사이에서 다수의 인스턴스들로 상기 명령들의 웨이크 및 실행을 야기하도록 동작가능하다.

대 표 도 - 도1



(52) CPC특허분류
G09G 5/14 (2013.01)
G09G 5/363 (2013.01)
G09G 5/399 (2013.01)
G09G 2310/061 (2013.01)
G09G 2310/08 (2013.01)
G09G 2340/0435 (2013.01)
G09G 2340/0435 (2013.01)
G09G 2360/08 (2013.01)
G09G 2360/18 (2013.01)

(56) 선행기술조사문헌
US20020030694 A1*
US20050083339 A1*
US20160247484 A1
US09251552 B
*는 심사관에 의하여 인용된 문헌

명 세 서

청구범위

청구항 1

시스템에 있어서,

디스플레이 디바이스(20) 상에 디스플레이하기 위해 비디오 컨텐츠를 렌더링하고 주기적으로 상기 디스플레이 디바이스를 리프레시하도록 동작가능한 컴퓨팅 디바이스(10)로, 상기 비디오 컨텐츠는 다수의 애플리케이션 윈도우들을 포함하는, 상기 컴퓨팅 디바이스(10);

상기 다수의 애플리케이션 윈도우들을 포함하는 합성된 외관들(surfaces)인 비디오 프레임들을 합성하고 상기 비디오 프레임들을 상기 디스플레이 디바이스로 전달하기 위해 버퍼 플립(flip)을 개시하는 명령들을 웨이크(wake)하고 실행하도록 동작가능한 데스크탑 합성자(90); 그리고

만일 렌더링된 비디오 컨텐츠가 이용가능하면 상기 데스크탑 합성자가 상기 명령들을 실행하기 위해 디스플레이 리프레시들 사이에서 다수의 인스턴스들(instances)로 웨이크를 야기하도록 동작가능한 고해상도 타이머(92)를 포함하는, 시스템.

청구항 2

청구항 1에 있어서, 상기 디스플레이 디바이스는 동적 리프레시 속도로 동작할 수 있는, 시스템.

청구항 3

청구항 1에 있어서, 상기 컴퓨팅 디바이스는 운영체제(80)를 포함하고, 상기 데스크탑 합성자는 상기 운영체제의 일부인, 시스템.

청구항 4

청구항 3에 있어서, 상기 고해상도 타이머는 상기 운영체제의 일부인, 시스템.

청구항 5

청구항 1에 있어서, 비디오 드라이버(85)를 포함하고, 상기 고해상도 타이머는 상기 비디오 드라이버의 일부인, 시스템.

청구항 6

청구항 1에 있어서, 상기 디스플레이 디바이스(20)는 수직 동기(vertical synchronization)(VSYNC) 타이밍이 인에이블된(enabled) 채로 동작하고, 상기 고해상도 타이머는 상기 데스크탑 합성자가 상기 VSYNC 타이밍과 독립적으로 상기 명령들의 웨이크 및 실행을 야기하도록 동작가능한, 시스템.

청구항 7

청구항 1에 있어서, 상기 고해상도 타이머는 버퍼가 렌더링을 위해 이용가능하다는 것을 다음 리프레시 전에 상기 데스크탑 합성자로 보고하도록 동작가능한, 시스템.

청구항 8

청구항 1에 있어서, 상기 고해상도 타이머는 상기 비디오 컨텐츠에 기초하여 동적으로 조정가능한 주파수들로 동작하도록 구성된, 시스템.

청구항 9

방법에 있어서,

디스플레이 디바이스(20) 상에 디스플레이하기 위해 비디오 컨텐츠를 렌더링하고 주기적으로 상기 디스플레이

디바이스를 리프레시하는 단계로서, 상기 비디오 컨텐츠는 다수의 애플리케이션 윈도우들을 포함하는, 상기 단계,

상기 다수의 애플리케이션 윈도우들을 포함하는 합성된 외관들인 비디오 프레임들을 합성하고 상기 비디오 프레임들을 상기 디스플레이 디바이스로 전달하기 위해 버퍼 플립을 개시하는 명령들을 웨이크하고 실행하기 위해 데스크탑 합성자(90)를 동작하는 단계; 그리고

만일 렌더링된 비디오 컨텐츠가 이용가능하면 상기 데스크탑 합성자가 상기 명령들을 실행하기 위해 디스플레이 리프레시들 사이에서 다수의 인스턴스들로 웨이크를 야기하도록 고해상도 타이머(92)를 동작하는 단계를 포함하는, 방법.

청구항 10

청구항 9에 있어서, 상기 디스플레이 디바이스를 동적 리프레시 속도로 동작하는 단계인, 방법.

청구항 11

청구항 9에 있어서, 상기 데스크탑 합성자는 운영체제(80)의 일부인, 방법.

청구항 12

청구항 11에 있어서, 상기 고해상도 타이머는 상기 운영체제의 일부인, 방법.

청구항 13

청구항 9에 있어서, 상기 고해상도 타이머는 비디오 드라이버(85)의 일부인, 방법.

청구항 14

청구항 9에 있어서, 상기 디스플레이 디바이스를 수직 동기(VSYNC) 타이밍이 인에이블된 채로 동작하고, 상기 고해상도 타이머는 상기 데스크탑 합성자가 상기 VSYNC 타이밍과 독립적으로 상기 명령들의 웨이크 및 실행을 야기하도록 동작하는 단계인, 방법.

청구항 15

청구항 9에 있어서, 상기 고해상도 타이머는 버퍼가 렌더링을 위해 이용가능하다는 것을 다음 리프레시 전에 상기 데스크탑 합성자로 보고하도록 동작하는 단계를 포함하는, 방법.

청구항 16

청구항 9에 있어서, 상기 고해상도 타이머는 상기 비디오 컨텐츠에 기초하여 동적으로 조정가능한 주파수들로 동작하는 단계를 포함하는, 방법.

청구항 17

디스플레이 디바이스(20)를 갖고 상기 디스플레이 디바이스 상에 디스플레이하기 위해 비디오 컨텐츠를 렌더링하고 주기적으로 상기 디스플레이 디바이스를 리프레시하도록 동작가능한 제1 컴퓨팅 디바이스(15)를 포함하는 시스템 내에서, 상기 비디오 컨텐츠는 다수의 애플리케이션 윈도우들을 포함하는, 방법으로서:

디스플레이 디바이스 상에 디스플레이하기 위해 상기 비디오 컨텐츠를 렌더링하고 주기적으로 상기 디스플레이 디바이스를 리프레시하는 단계;

데스크탑 합성자(90)가 상기 다수의 애플리케이션 윈도우들을 포함하는 합성된 외관들인 비디오 프레임들을 합성하고 상기 비디오 프레임들을 상기 디스플레이 디바이스로 전달하기 위해 버퍼 플립을 개시하는 명령들을 웨이크하고 실행하도록 동작하는 단계; 그리고

만일 렌더링된 비디오 컨텐츠가 이용가능하면 상기 데스크탑 합성자가 상기 명령들을 실행하기 위해 디스플레이 리프레시들 사이에서 다수의 인스턴스들로 웨이크를 야기하도록 고해상도 타이머(92)를 동작하는 단계를 포함하는, 방법.

청구항 18

청구항 17에 있어서, 상기 디스플레이 디바이스를 동적 리프레시 속도로 동작하는 단계인, 방법.

청구항 19

청구항 17에 있어서, 상기 데스크탑 합성자는 운영체제(80)의 일부인, 방법.

청구항 20

청구항 19에 있어서, 상기 고해상도 타이머는 상기 운영체제의 일부인, 방법.

청구항 21

청구항 17에 있어서, 상기 고해상도 타이머는 비디오 드라이버(85)의 일부인, 방법.

청구항 22

청구항 17에 있어서, 상기 디스플레이 디바이스를 수직 동기(VSYNC) 타이밍이 인에이블된 채로 동작하고, 상기 고해상도 타이머는 상기 데스크탑 합성자가 상기 VSYNC 타이밍과 독립적으로 상기 명령들의 웨이크 및 실행을 야기하도록 동작하는 단계인, 방법.

청구항 23

청구항 17에 있어서, 상기 고해상도 타이머는 버퍼가 렌더링을 위해 이용가능하다는 것을 다음 리프레시 전에 상기 데스크탑 합성자로 보고하도록 동작하는 단계를 포함하는, 방법.

청구항 24

디스플레이 디바이스(20) 상에 디스플레이하기 위해 비디오 컨텐츠를 렌더링하고 주기적으로 상기 디스플레이 디바이스를 리프레시하는 단계, 상기 비디오 컨텐츠는 다수의 애플리케이션 윈도우들을 포함하는, 상기 단계,

상기 다수의 애플리케이션 윈도우들을 포함하는 합성된 외관들인 비디오 프레임들을 합성하고 상기 비디오 프레임들을 상기 디스플레이 디바이스로 전달하기 위해 버퍼 플립을 개시하는 명령들을 웨이크하고 실행하기 위해 데스크탑 합성자(90)를 동작하는 단계; 그리고

만일 렌더링된 비디오 컨텐츠가 이용가능하면 상기 데스크탑 합성자가 상기 명령들을 실행하기 위해 디스플레이 리프레시들 사이에서 다수의 인스턴스들로 웨이크를 야기하도록 고해상도 타이머(92)를 동작하는 단계를 포함하는 방법을 수행하기 위해 컴퓨터 판독 가능한 명령들을 갖는 비-일시적 컴퓨터 판독가능 매체.

발명의 설명

기 술 분 야

[0001] 본 출원은 2016년 10월 31일자로 출원된 미국 특허 출원 번호 15/338,492의 우선권을 주장하며, 그 전체 내용은 본 명세서에 참조로 통합된다.

[0002] 본 발명은 일반적으로 컴퓨팅 시스템들 및 소프트웨어에 관한 것으로, 보다 상세하게는 그래픽 컨텐츠를 디스플레이하기 위한 컴퓨팅 시스템들 및 소프트웨어에 관한 것이다.

배 경 기 술

[0003] 대부분의 현대 컴퓨터 운영체제는 전체화면(fullscreen) 모드 (또는 심지어 전체화면 전용 모드) 또는 윈도우(windowed) 모드에서 애플리케이션을 디스플레이할 능력을 포함한다. 몇몇 종래의 컴퓨터 모니터들은 고정된 리프레시 속도(refresh rate)를 갖고 다른 것들은 동적으로 리프레시될 수 있다. 종래의 컴퓨터 시스템들에서, 모니터 상에 디스플레이할 컨텐츠는 먼저 그래픽 처리 유닛(GPU)과 같은 렌더링 하드웨어에 의해 렌더링 또는 생성되고, 그리고는 전형적으로 시스템 메모리 또는 GPU 메모리/캐시들 내의 주소 위치인 두 개의 버퍼들 중 하나에 저장된다. 많은 종래의 시스템들은 두 개의 버퍼들을 소위 더블 버퍼링 스킴(double buffering scheme) 내에서 사용한다. 전형적으로 프론트 버퍼(front buffer)로 참조되는 하나의 버퍼는 모니터 상에 현재 디스플레이되는 렌더링된 컨텐츠를 보유한다. 전형적으로 백 버퍼(back buffer)인 다른 버퍼는 다음의 적절한 시간에 디스플레이되기를 기다리는 렌더링된 컨텐츠를 보유한다. 운영체제가 백 버퍼 컨텐츠가 디스플레이 되어야 한다고

결정하면, 디스플레이가 백 버퍼의 컨텐츠를 디스플레이하는 것으로 스위칭하는 플립(flip)이 수행된다.

[0004] 윈도우 모드에서, 렌더링 프로세스는 데스크탑의 비어있는 임의의 부분과 작업표시줄과 메뉴들과 함께 각각의 열려 있는 애플리케이션의 컨텐츠를 렌더링한 후에 이들 개별적인 그래픽 부분들을 디스플레이 상에 프레임 단위로 (frame after frame) 디스플레이되는 합성된 외관(surface)으로 합성하는 것을 포함한다. 후자의 프로세스는 데스크탑 합성이라고 알려져 있고 일반적으로 데스크탑 합성자(compositor)라고 불리우는 운영체제 프로세스에 의해 수행된다. 데스크탑 합성자는 복수개의 애플리케이션 윈도우들을 스캔 아웃(scan out)되어 모니터 상에 디스플레이되는 단일 외관으로 합성하기 위해 전형적으로 모니터의 리프레시 속도와 관련있는 고정된 간격으로 웨이크되는(wake up) 시스템 프로세스이다.

[0005] 종래의 운영 체제는 상이한 컨텐츠들을 갖는 두 개의 렌더링된 프레임들의 부분들을 디스플레이가 보여주는 "티어링(tearing)"이라고 알려진 현상을 피하기 위해 수직 동기 또는 VSYNC라고 불리우는 프로세스를 종종 사용한다. VSYNC가 인에이블되면(enabled), 각각의 버퍼 플립은 각각의 리프레시 이후에만 발생한다. 이것은 프레임 렌더링 속도를 모니터 리프레시 속도로 효과적으로 한정한다. VSYNC는 또한 데스크탑 합성자가 웨이크될 때 고정된 간격을 설정한다.

[0006] 전체 화면 전용 모드에서 실행되는 현재의 애플리케이션들은 애플리케이션이 그 렌더링을 완료할 때로부터 컨텐츠가 동적 리프레시 속도 기술로 모니터 상에 나타날 때까지 사이에 아주 적은 대기시간 (<1 ms)을 갖는다. 이것은 애플리케이션이 백 버퍼로의 플립을 직접 제어하고 그리고는 렌더링이 완료된 후 즉시 플립을 개시하는 사실에 기인한다. 하지만 데스크탑 합성의 추가적인 단계를 갖는 윈도우가 있는 애플리케이션에 대해서는, 대기시간이 매우 높고 전형적인 60Hz 디스플레이에 대해 잠재적으로 33.3ms에 이를 수도 있다. 이러한 지연은 다음을 포함한다: (1) 데스크탑 합성자 스레드가 웨이크되는 것을 기다리는데 사용된 시간; 및 (2) 렌더링 하드웨어가 프론트 및 백 버퍼들을 플립할 수 있기 전에 VSYNC 바운더리 (다음의 고정된 리프레시)을 기다리는데 사용된 시간. 이것은 눈에 띌만한 시각적인 버벅거림(stuttering)을 초래한다. 이러한 대기시간을 최소화하거나 적어도 감소시킬 수 있다면, 윈도우가 있는 애플리케이션들은 전체화면 전용 모드 애플리케이션의 성능에 가까운 성능을 수행할 수 있을 것이고, 동적 리프레시 속도 모니터들 상에서 낮은 대기시간 및 버벅거림-없는 시각적인 경험을 성취할 수 있을 것이다.

[0007] Windows® 10 은 종래의 전체화면 전용의 낮은 대기시간의 이점을 갖고 또한 빠른 태스크 전환의 편리함을 갖는 에뮬레이트된 전체화면 전용 모드를 소개하였다. 하지만, 이러한 구현은 윈도우가 있는 애플리케이션들 모두를 커버하지 않고, 또한 GPU가 두 개의 완전히 다른 외관 포맷들 (데스크탑 합성자 외관 대 애플리케이션 외관) 사이를 매끄럽게 플립할 수 있는지에 기초한 제한이 있다.

[0008] 또 다른 종래 기술에서, 사용자는 낮은 대기시간 경험을 위해 전체화면 전용 모드로 직접 실행할 것을 선택할 수 있다. 하지만 문제점은 애플리케이션들 사이에서, 특히 사용자가 상이한 모니터들 상에서 실행되는 애플리케이션들을 사용하는 것들 사이에서 전환할 수 있는 다중-디스플레이 환경 설정에서 애플리케이션들 사이에서 포커스(focus)가 전환될 때의 긴 태스크 전환 시간을 포함한다. 다른 문제점은 일부 애플리케이션들은 전체화면 전용 모드에서 수행되는 옵션을 전혀 갖고 있지 않다는 것이다. 추가적으로, 데스크탑 합성자는 음량 또는 밝기 조절과 같은 유용한 컨텐츠를 스크린 상에 오버레이 할 수 없다.

[0009] 본 발명은 전술한 단점들의 하나 이상의 효과를 극복 또는 감소시키는 것에 관한 것이다.

발명의 내용

[0010] 본 발명의 일 측면에 따르면, 디스플레이 디바이스 상에 디스플레이하기 위해 비디오 컨텐츠를 렌더링하고 주기적으로 상기 디스플레이 디바이스를 리프레시하도록 동작가능한 컴퓨팅 디바이스를 포함하는 시스템이 제공된다. 상기 비디오 컨텐츠는 적어도 하나의 애플리케이션 윈도우를 포함한다. 데스크탑 합성자는 상기 적어도 하나의 애플리케이션 윈도우를 포함하는 합성된 외관들인 비디오 프레임들을 합성하고 상기 비디오 프레임들을 상기 디스플레이 디바이스로 전달하기 위해 버퍼 플립을 개시하는 명령들을 웨이크하고 실행하도록 동작가능하다. 고해상도 타이머는 상기 데스크탑 합성자가 디스플레이 리프레시들 사이에서 다수의 인스턴스들로 상기 명령들의 웨이크 및 실행을 야기하도록 동작가능하다.

[0011] 본 발명의 다른 측면에 따르면, 디스플레이 디바이스 상에 디스플레이하기 위해 비디오 컨텐츠를 렌더링하고 주기적으로 상기 디스플레이 디바이스를 리프레시하는 방법이 제공된다. 상기 비디오 컨텐츠는 적어도 하나의 애플리케이션 윈도우를 포함한다. 데스크탑 합성자는 상기 적어도 하나의 애플리케이션 윈도우를 포함하는 합성된

외관들인 비디오 프레임들을 합성하고 상기 비디오 프레임들을 상기 디스플레이 디바이스로 전달하기 위해 버퍼 플립을 개시하는 명령들을 웨이크하고 실행하기 위해 동작된다. 고해상도 타이머는 상기 데스크탑 합성자가 디스플레이 리프레시들 사이에서 다수의 인스턴스들로 상기 명령들의 웨이크 및 실행을 야기하도록 동작된다.

[0012] 본 발명의 또 다른 측면에 따르면, 디스플레이 디바이스를 갖고 상기 디스플레이 디바이스 상에 디스플레이하기 위해 비디오 컨텐츠를 렌더링하고 주기적으로 상기 디스플레이 디바이스를 리프레시하도록 동작가능한 컴퓨팅 디바이스를 포함하는 시스템 내에서, 상기 비디오 컨텐츠는 적어도 하나의 애플리케이션 윈도우를 포함하고, 방법은 디스플레이 디바이스 상에 디스플레이하기 위해 상기 비디오 컨텐츠를 렌더링하고 주기적으로 상기 디스플레이 디바이스를 리프레시하는 것을 포함한다. 데스크탑 합성자는 상기 적어도 하나의 애플리케이션 윈도우를 포함하는 합성된 외관들인 비디오 프레임들을 합성하고 상기 비디오 프레임들을 상기 디스플레이 디바이스로 전달하기 위해 버퍼 플립을 개시하는 명령들을 웨이크하고 실행하도록 동작된다. 고해상도 타이머는 상기 데스크탑 합성자가 디스플레이 리프레시들 사이에서 다수의 인스턴스들로 상기 명령들의 웨이크 및 실행을 야기하도록 동작된다.

[0013] 본 발명의 또 다른 측면에 따르면, 비-일시적 컴퓨터 판독 가능한 매체는 방법을 수행하기 위한 컴퓨터 판독 가능한 명령들을 갖는다. 상기 방법은 디스플레이 디바이스 상에 디스플레이하기 위해 비디오 컨텐츠를 렌더링하고 주기적으로 상기 디스플레이 디바이스를 리프레시하는 것을 포함한다. 상기 비디오 컨텐츠는 적어도 하나의 애플리케이션 윈도우를 포함한다. 데스크탑 합성자는 상기 적어도 하나의 애플리케이션 윈도우를 포함하는 합성된 외관들인 비디오 프레임들을 합성하고 상기 비디오 프레임들을 상기 디스플레이 디바이스로 전달하기 위해 버퍼 플립을 개시하는 명령들을 웨이크하고 실행하도록 동작된다. 고해상도 타이머는 상기 데스크탑 합성자가 디스플레이 리프레시들 사이에서 다수의 인스턴스들로 상기 명령들의 웨이크 및 실행을 야기하도록 동작된다.

도면의 간단한 설명

[0014] 전술한 그리고 본 발명의 다른 이점들은 후술할 상세한 설명을 읽고 다음의 도면들을 참조하여 명확해질 것이다:

도 1은 컴퓨터 및 디스플레이 또는 모니터를 포함하는 컴퓨팅 시스템의 예시적인 실시예의 그림으로 된 도면이고;

도 2는 예시적인 컴퓨팅 시스템의 블록도이고;

도 3은 도 1과 같은, 하지만 디스플레이 상에 단일 윈도우모드 애플리케이션을 도시한 그림으로 된 도면이고;

도 4는 예시적인 종래의 비디오 렌더링 및 버퍼 플리핑 방법을 도시한 활동도이고;

도 5는 도 4와 같은, 하지만 간략화시킨 예시적인 종래의 비디오 렌더링 및 버퍼 플리핑 방법을 도시한 활동도이고;

도 6은 도 5와 같은, 하지만 새로운 비디오 렌더링 및 버퍼 플리핑 방법의 예시적인 실시예를 도시한 활동도이고;

도 7은 도 5와 같은, 하지만 또 다른 간략화시킨 예시적인 종래의 비디오 렌더링 및 버퍼 플리핑 방법을 도시한 활동도이고;

도 8은 도 7과 같은, 하지만 새로운 비디오 렌더링 및 버퍼 플리핑 방법의 또 다른 예시적인 실시예를 도시한 활동도이고;

도 9는 도 7과 같은, 하지만 간략화시킨 예시적인 종래의 비디오 렌더링 및 버퍼 플리핑 방법의 추가적인 측면들을 도시한 활동도이고;

도 10은 도 8과 같은, 하지만 새로운 비디오 렌더링 및 버퍼 플리핑 방법의 예시적인 실시예의 추가적인 측면들을 도시한 활동도이고;

도 11은 감소된 플립 대기시간을 갖는 비디오 컨텐츠를 렌더링하는 예시적인 방법을 도시하는 흐름도이고; 그리고

도 12는 동적 타이머 주파수가 인에이블된 또는 인에이블되지 않은 예시적인 렌더링을 도시한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0015] 윈도우 모드에서 애플리케이션들에게 비디오를 렌더링하도록 동작가능한 컴퓨팅 시스템들의 다양한 실시예들이 개시된다. 일 변형은 디스플레이 디바이스 상에서 디스플레이하기 위해 비디오 컨텐츠를 렌더링하고 해당 디스플레이 디바이스를 주기적으로 리프레시하도록 동작가능하다. 비디오 컨텐츠는 적어도 하나의 애플리케이션 윈도우를 포함한다. 운영체제 프로세스일 수도 있는 데스크탑 합성자는 웨이크되고 적어도 하나의 애플리케이션 윈도우를 포함하는 합성된 외관들인 비디오 프레임들을 합성하고 디스플레이 디바이스로 비디오 프레임들을 전송하도록 버퍼 플립을 개시하는 명령들을 실행하도록 동작가능하다. 고해상도 타이머는 데스크탑 합성자가 디스플레이 리프레시들 사이의 다수의 인스턴스들로 명령들을 웨이크되고 실행하도록 동작가능하다. 종래의 시스템들에서, 데스크탑 합성자는 각 리프레시 사이클 후에에만 한번 웨이크되어 실행하기만 하며, 이는 버퍼 플리핑 시 지연이 생기게 한다. 추가적인 상세 내용은 이제 설명될 것이다.

[0016] 이하에 설명되는 도면들에서, 도면 부호들은 동일한 요소가 하나 이상의 도면에 나타날 때 일반적으로 반복된다. 이제 도면을 보면, 특히 도 1에서, 컴퓨터(15) 및 컴퓨터(15)에 연결된 디스플레이 또는 모니터(20)을 포함하는 컴퓨팅 시스템(10)의 예시적인 실시예의 그림으로 된 도면이 도시된다. 대안적인 실시예들에서, 컴퓨터(15) 및 디스플레이(20)는 하나의 케이스로 통합될 수 있다. 디스플레이(20)는 LCD, OLED, 플라즈마 또는 다른 기술들과 같은 다양한 기술들 중 임의의 것일 수 있다. 디스플레이(20)는 유리하게는 이하에 더욱 상세히 설명되는 바와 같이 동적 리프레시를 할 수 있다. 컴퓨터 시스템(10)은 예시에서의 간략화를 위해 보여지지 않는 마우스들 및 키보드들과 같은 다양한 입력 디바이스들을 포함할 수 있다.

[0017] 컴퓨터(15)는 Windows®, Linux, IOs 또는 다른 것들과 같은 하나 이상의 운영체제들을 이용할 수 있다. 물론, 여기에서 기재된 기술들은 많은 방식들에 있어 플랫폼에 구속되지 않는다. 본 예시적인 실시예에서, 운영체제 소프트웨어는 데스크탑(25), 열려 있는 두 개의 예시적인 윈도우가 있는 애플리케이션(30 및 35) 및 몇몇 숫자의 아이콘들(45)을 포함하는 작업표시줄(40)을 포함하는 외관(22) 또는 컨텐츠를 디스플레이(20) 상에 디스플레이하고 있다. 애플리케이션들(30 및 35)은 물론 둘 이상일 수 있고 물론 하나 및 몇몇 정해지지 않은 숫자들 사이일 수 있다. 더욱 상세히 후술하는 바와 같이, 데스크탑(25) 및 열려 있는 윈도우가 있는 애플리케이션들(30 및 35)의 결합은 동적으로 리프레시되는 합성된 외관(22)을 이룬다.

[0018] 컴퓨팅 시스템(10)의 추가적인 상세 내용은 블록도인 도 2를 또한 참조함으로써 이해될 수 있다. 컴퓨팅 디바이스(10)는 전술한 디스플레이(20), 프로세서(50) 또는 프로세서들, 저장 디바이스(55), 매체(60) 및 메모리(65)를 포함할 수 있다. 프로세서(50)는 비디오 처리 전용 집적 회로, 마이크로프로세서, 그래픽 처리 유닛(GPU), 마이크로프로세서 및 그래픽 프로세서 기능들을 결합한 가속 처리 장치(APU), 애플리케이션 특정 집적 회로 또는 다른 디바이스일 수 있다. 예시적인 APU는 압축, 합축해지, 사전-부과된(pre-imposed) 또는 사후-부과된(post-imposed) 처리 태스크들 또는 다른 것들에 대한 고정된 기능 코어들을 포함할 수 있다. 물론, 프로세서(50)는 병렬로 또는 다르게 동작하는 그러한 집적 회로들의 다수의 예들로 구성될 수 있다. 저장 디바이스(55)는 컴퓨터 판독 가능한 매체이고 임의의 종류의 하드 디스크, 광학 저장 디스크, 고체 상태 저장 디바이스, ROM, RAM, 또는 컴퓨터 판독 가능한 매체를 저장하기 위한 가상적인 임의의 다른 시스템일 수 있다. 매체(60)는 위성 튜너, 케이블 셋탑 박스, 광학 디스크 플레이어, 인터넷 스트리밍, 제거 가능한 저장 디바이스 또는 하드 드라이브에 의해 공급되는 매체를 포함할 수 있다. 이들은 비디오 컨텐츠를 프로세서(50)로, 그러므로 도 1에 도시된 디스플레이(20)로 전달하는데 사용될 수 있는 매체들의 유형들의 단지 몇 가지 예들을 나타낸다. 메모리(65)는 DRAM, SRAM, VRAM 또는 플래시 칩들, 보드들 또는 모듈들 또는 온보드 캐시(들) 또는 이들의 결합과 같이 하나 이상의 개별적인 메모리 디바이스일 수 있다. 메모리(65)는 다양한 것들 중에서도 복수의 버퍼들, 버퍼1, 버퍼2, ... 버퍼m으로 세분화될 수 있다. 버퍼들 버퍼1, 버퍼2, ... 버퍼m은 이하에 더욱 상세히 설명되는 바와 같이 비디오 프레임들을 디스플레이하기 위한 더블, 트리플 또는 다른 버퍼링을 위해 사용될 수 있는 주소를 갖는 메모리 위치들이다.

[0019] 추가적으로, 컴퓨팅 디바이스(10)는 운영체제(80), 비디오 드라이버 소프트웨어(85), 데스크탑 합성자(90), 고해상도 타이머(92)를 포함하고, 그리고 앱1, 앱2, ... 앱n으로 줄여쓸 수 있고 드라이버들, 소프트웨어 애플리케이션들 또는 다른 형태의 애플리케이션들일 수 있는 복수개의 애플리케이션들을 포함할 수 있다. 예를 들어, 앱1은 열릴때 윈도우가 있는 애플리케이션(30)에 대응할 수 있고 앱2는 열릴때 도 1에서 보여지는 윈도우가 있는 애플리케이션(35)에 대응할 수 있다. 운영체제(80), 비디오 드라이버 소프트웨어(85), 고해상도 타이머(90), 데스크탑 합성자(90) 및 애플리케이션들 앱1 ... 앱n은 저장 디바이스(55)에 저장될 수 있다. 전술된 바와 같이, 운영체제 소프트웨어(80)는 매우 다양한 서로 다른 운영 체제들 중 임의의 것일 수 있다. 비디오 드라이버 소프트웨어(85)는 유사하게 다양한 서로 다른 운영체제 플랫폼들에 대해 적합할 수 있다.

[0020] 데스크탑 합성자(90) (또는 DWM 프로세스란 용어를 갖는)는 보여지는 바와 같이 운영체제(80)의 일부이고 시스템 프로세스로서 기능할 수 있다. 버퍼들 퍼버1, 버퍼2 ... 버퍼m 중 하나 이상(그리고 더블 버퍼링의 경우 두 개)은 렌더링된 애플리케이션 컨텐츠를 저장하기 위해 애플리케이션들 앱1, 앱2 ... 앱n 각각에 의해 할당될 수 있다. 추가적으로, 하나 이상의 버퍼들 (그리고 더블 버퍼링인 경우 두 개)버퍼1, 버퍼2 ... 버퍼m은 합성된 렌더링된 컨텐츠를 저장하기 위해 데스크탑 합성자(90)에 의해 할당될 수 있다.

[0021] 아래에 더욱 상세히 설명되는 바와 같이, 고해상도 타이머(92)는 데스크탑 합성자(90)가 예를 들어 데스크탑(25) 및 도 1에 도시된 열려 있는 애플리케이션들(30 및 35)로 구성된 합성인 데스크탑 외관(22) 을 생성하는 방식을 데스크탑 합성 및 디스플레이(20) 상에서 데스크탑 합성된 외관(22)의 실제 디스플레이를 생성하는데 요구되는 다양한 동작들 사이에서 대기시간을 감소시키는 기술적인 목표를 갖고 수정하도록 디자인될 수 있다. 고해상도 타이머(92)는 비디오 드라이버 소프트웨어(85), 운영체제 소프트웨어(80) 또는 아마도 저장 디바이스(55) 또는 다른 위치 내에서 심지어 펌웨어로서 구현될 수도 있다. 만일 펌웨어로 구현되면, 고해상도 타이머(92)는 원하는 기능을 구현하기 위해 비디오 드라이버 소프트웨어(85)를 통해 유리하게 동작해야 한다.

[0022] 데스크탑 합성자(90)를 이용하여 비디오를 렌더링하기 위한 예시적인 종래의 기술은 이제 도 2, 3 및 4를 참조하여 이해될 수 있다. 도 3은 초기에 데스크탑(25) 및 윈도우가 있는 애플리케이션(35)만을, 즉 데스크탑(25) 및 윈도우가 있는 애플리케이션(35)으로 구성된 데스크탑 외관(22)을 디스플레이하는 컴퓨팅 시스템(10)의 그림으로 된 도면이다. 도 4는 도 3에 도시된 데스크탑 외관(22)을 디스플레이하는 것과 연관된 다양한 하드웨어 및 소프트웨어 활동들을 도시하는 타이밍도에 유사한 활동도이다. 위에서부터 밑으로 가면서, 도 4는 하드웨어 큐(Hardware Queue) 스트림(100), 데스크탑 합성자(DWM) 프로세스 스트림(105), 애플리케이션 생성 컨텐츠 스트림(110), 계류중인 플립 스트림(115) 및 온스크린 디스플레이 스트림(120)를 도시한다. 하드웨어 큐 스트림(100)은 컨텐츠의 하드웨어 렌더링을 나타낸다. 렌더링된 컨텐츠는 도 3에서 윈도우가 있는 애플리케이션(35)의 컨텐츠의 렌더링과 같은 애플리케이션 생성 컨텐츠, 그리고 도 3에서 합성된 외관(22)을 렌더링하는 것과 같은 합성된 컨텐츠의 렌더링일 수 있다. 렌더링은 실제로 비디오 프레임을 기입하고 있는 임의의 하드웨어에 의해 수행된다. 이것은 도 2에 도시된 프로세서(50) 또는 만일 GPU가 컴퓨팅 시스템(10) 내에 구현되면 별개의 GPU일 수도 있으며, 이들 둘이 협력하거나 또는 몇몇 다른 디바이스(들)일 수도 있다. DWM 프로세스 스트림(105)은 이하에 더욱 상세히 설명될 다양한 기능들을 수행하기 위해 도 2에 보여지는 운영체제 소프트웨어(80) 내에서의 DWM 프로세스(즉, 데스크탑 합성자(90))의 주기적인 웨이크를 도시한다. 애플리케이션 생성 컨텐츠 스트림(110)은 특정 애플리케이션에 의해 생성된 새로운 컨텐츠, 예컨대 컨텐츠(A),를 도시한다. 이것은, 예를 들어, 디스플레이(20)상에 렌더링될 도 3에서 보여지는 윈도우가 있는 애플리케이션(35)을 위한 일부 업데이트된 비디오 정보일 수 있다. 계류중인 플립 스트림(115)은 데스크탑 합성자(90)가 언제 두 개의 DWM 할당된 버퍼들 사이에서 버퍼 플립, 말하자면 버퍼1로부터 버퍼2로 (그리고 궁극적으로는 디스플레이(20)로의 전달을 위해),을 개시하는지 그리고 버퍼1 및 버퍼2 사이의 플립에서 지연, 도시된 지연들(125 및 130)과 같은, 이 있는지를 나타낸다. 이러한 지연들 (125 및 130)은 이하에 보다 상세하게 설명될 것이다. 마지막으로, 온스크린 디스플레이 스트림(120)은 현재 디스플레이(20) 상에 실제로 디스플레이되는 것을 도시한다. 예를 들어, 버퍼1의 컨텐츠는 초기에 디스플레이(20) 상에 디스플레이되고 있고, 버퍼1은 프로세서(50) 또는 보드상에 있는 임의의 프로세서로부터 렌더링되고 전달된 데스크탑 외관(22)을 저장하고 있을 수 있다.

[0023] 계속하여 도 4를 참조하면, 주기적인 리프레시 선들, 리프레시1, 리프레시2, 리프레시3, 리프레시4, 리프레시5 및 리프레시6이 도시된 것이 주목된다. 이들은 디스플레이(20)의 개별적인 리프레시들을 나타낸다. 종래의 시스템에서 그리고 도 4에 도시된 바와 같이, 리프레시1 및 리프레시2 등은 VSYNC라고 이름붙여진 종래의 수직 동기화 코드에 의해 설정된 시간에서 특정한 고정 시점에 나타난다. VSYNC가 인에이블되면, 프레임 렌더링 속도는 모니터 리프레시 속도로 효과적으로 한정된다. DWM 프로세스 스트림(105)에서 보여지는 바와 같이, DWM 프로세스 웨이크/실행의 인스턴스는 각각의 리프레시, 즉 리프레시1, 리프레시2 등 이후 즉시 발생한다. 예를 들어, 리프레시1 후, DWM 프로세스 웨이크/실행 인스턴스(135)가 발생한다. 이 시점에서 애플리케이션이 컨텐츠(A)를 생성하고 있고, 이는 다시 단순히 도 3에서 도시된 애플리케이션(35)을 위한 업데이트된 프레임일 수 있다. 온스크린 스트림(120)에 의해 보여지는 바와 같이, 버퍼1의 컨텐츠, 도 3에서 보여지는 데스크탑 외관(22)인,가 디스플레이되고 있다. 후술될 이유들로 인해 리프레시1 후 계류중인 플립 스트림(115)에서 보여지는 바와 같이 계류중인 버퍼 플립은 없다. DWM 프로세스 웨이크/실행 인스턴스(135)는 이용가능한 렌더링된 컨텐츠를 찾고, 만일 렌더링된 컨텐츠가 이용가능하면, 비디오 드라이버(85)에게 이용가능한 컨텐츠의 주소 위치(버퍼m일 수 있다)를 알린다. DWM 프로세스 웨이크/실행 인스턴스(135)는 그리고는 닫힌다. 비디오 드라이버(85)는 그리고는 새로운 버퍼 주소(말하자면 버퍼m)를 프로세서(50)(또는 온보드인 경우 GPU)로 프로그램하고, 그리고 프로세서

는 다음 리프레시를 개시한다. 만일 렌더링된 컨텐츠가 감지되지 않으면, DWM 프로세스 웨이크/실행 인스턴스(135)는 닫히고 다음 리프레시를 기다린다. 리프레시2에 이어 하지만 리프레시3 이전에, 다음 DWM 프로세스 웨이크/실행 인스턴스(140)가 발생하고 이 기간 동안 애플리케이션 생성 컨텐츠(A)가 완료되고 업데이트된 컨텐츠(A)가 하드웨어 큐 스트림(100)에서 보여지는 바와 같이 렌더링을 완료한다. 이 렌더링된 컨텐츠(A)는 애플리케이션 할당된 버퍼, 예컨대 버퍼m,에 저장될 것이다. 하지만, DWM 프로세스 웨이크/실행 인스턴스(140)가 이미 열렸고 컨텐츠(A)의 렌더링의 완료 전에 닫혔으므로, 대기가 먼저 발생하게 된다. 대기의 3가지 잠재적 원인들이 있다: 합성시 지연, 하드웨어 지연 및 계류중인 플립 지연. 이들 원인들은 추가적이고 이하에 더욱 상술할 것이다. 예를 들어 도 2에서 보여지는 데스크탑 합성자(90)를 이용하여 DWM 합성을 이 시점에서 개시하여 DWM 버퍼1로부터 버퍼2로의 플립이 발생할 수 있도록 하는 것이 유리할 것이다. 하지만, 합성 프로세스는 발생할 수 없는데 이는 DWM 프로세스 인스턴스(140)가 이미 발생했고 다음 것은 리프레시3까지 발생하지 않기 때문이다. 따라서, 온스크린 컨텐츠는 여전히 버퍼1이다. 리프레시3에 이어, 다음 DWM 프로세스 웨이크/실행 인스턴스(145)가 발생한다. 이 시점에서, 렌더링된(A) 컨텐츠는 애플리케이션 할당된 버퍼인 버퍼m에 로드되었고 하드웨어 큐 스트림(100)이 나타내듯이 렌더링 하드웨어는 애플리케이션 생성 컨텐츠(B)를 렌더링하기 시작하였고, 이것은 다시 완전히 새로운 컨텐츠 또는 단순히 업데이트된 프레임일 수 있다. 하드웨어 큐 스트림(100)에서 렌더(A) 시점의 끝에서 발생할 수 있는 DWM 합성 프로세스는 적어도 DWM 프로세스 웨이크/실행 인스턴스(145)가 발생하는 리프레시3까지 지연되는 것이 주목된다. 리프레시3 이후 즉시 하드웨어 지연이 없다면, 하드웨어 큐(100)에서 보여지는 DWM 합성 프로세스(150)는 즉시 발생할 수 있다. 하지만, 도 4는 렌더링 컨텐츠(B) 또는 다양한 다른 요인들과 연관된 복잡성에 의해 야기될 수 있는 전형적이고 예시적인 하드웨어 지연을 도시한다. 그러므로, 도시된 예에서, DWM 합성 프로세스(150)는 리프레시4 전 짧게 얼마간의 시간동안까지는 완료되지 않는다. DWM 프로세스 웨이크/실행 인스턴스(145)는 애플리케이션 컨텐츠(A)의 이용가능성을 감지하고, 화살표(152)에 의해 나타내지듯이, 렌더링 하드웨어가 하드웨어 큐(100)에서 반영된 바와 같이 DWM 합성 프로세스(150)를 수행하도록 한다. DWM 합성 프로세스(150)는 데스크탑(25) 및 윈도우가 있는 애플리케이션(35)(애플리케이션 할당된 버퍼인 버퍼m에 현재 저장된)의 컨텐츠를 취하고, 이들 두개로부터 이들 둘의 합성인 데스크탑 외관(22)을 합성하고, 그리고는 그 합성된 데스크탑 외관(22)을 다른 DWM 할당된 버퍼인 버퍼2에 저장하는 것을 포함한다. DWM 합성 프로세스(150)의 끝에서, 도 3에서 보여지는 데스크탑 외관(22)의 컨텐츠는 버퍼2에 기입된다. 하지만, 버퍼1로부터 버퍼2로의 플립은 다음 리프레시, 즉 리프레시4까지 발생할 수 없다. 따라서, 플립 지연(125)가 발생하게 된다. 리프레시4 이전에, 온스크린 컨텐츠는 여전히 버퍼1이나 애플리케이션이 애플리케이션 생성 컨텐츠(C)를 생성하였고 하드웨어 큐(100)가 이제 애플리케이션 생성 컨텐츠(C)의 렌더링을 나타내는 것이 주목된다. 리프레시4에서, 버퍼2로의 플립이 발생하고 그러므로 온스크린 디스플레이 스트림(120)은 버퍼의 컨텐츠가 이제 디스플레이(20)상에 있다는 것을 반영한다. 리프레시2 및 리프레시4 사이의 시간대를 되돌아보면, 대기시간의 세가지 가능한 요소들, 하드웨어 큐(100)에서 컨텐츠(A)의 렌더링 및 다음 리프레시인 리프레시3 사이의 합성시 지연, 뿐만 아니라 리프레시3 및 DWM 합성 프로세스(150)의 끝 사이의 전형적인 하드웨어 지연 그리고 마지막으로 DWM 합성 프로세스(150)의 끝 및 다음 리프레시인 리프레시4 사이의 플립 지연(125),이 있다는 것이 명백하다.

[0024] 계속 도 4를 참조하면, 리프레시4 및 리프레시5 사이에서, 애플리케이션 생성 컨텐츠는 계속해서 컨텐츠(C)이고 (C)의 렌더링은 하드웨어 큐(100)에서 보여지는 바와 같이 계속된다. 리프레시5에 이어, DWM 프로세스 웨이크/실행 인스턴스(155)는 애플리케이션 컨텐츠(B)의 이용가능성(즉, 렌더(B)가 완료됨)을 감지하고, 다시 화살표(152)에 의해 나타내지는 바와 같이, 렌더링 하드웨어가 하드웨어 큐(100)에 반영된 것과 같이 다음 DWM 합성 프로세스(160)를 수행하도록 한다. 다음 DWM 합성 프로세스(160)의 끝에서, 합성된 컨텐츠 프레임이 버퍼1에 기입된다. 하지만, VSYNC 부과된 리프레시 시점들 때문에, 리프레시6인 다음 리프레시시까지 버퍼2로부터 다시 버퍼1로 컨텐츠의 플리핑은 발생할 수 없고 그러므로 전술한 플립 지연(130)이 발생한다.

[0025] 개시된 실시예들은 부분적으로는 도 4에 단지 설명되고 도시된 지연의 원인들 중 일부를 제거하는 것에 관한 것이다. 개시된 실시예들에 따른 예시적인 새로운 기술을 설명하기 전에, 먼저 도 3을 이용하여, 하지만 이제 또한 도 4와 같은 활동도이나 컨텐츠(A)의 렌더링 및 오직 4개의 리프레시 사이클들만을 포함하도록 간략화된 도 5를 이용하여 간략화된 프로세스 흐름을 다시 설명하는 것이 유용할 수 있다. 종래의 프로세스들에 대한 설명에 이어, 도 6 및 동일한 컨텐츠를 렌더링하기 위한 그리고 도 2에서 도시되고 위에서 간략히 설명된 고해상도 타이머(92)를 이용함으로써 예시적인 새로운 프로세스에 대해 참조가 행해질 것이다.

[0026] 도 5는 도 4와 같이 하드웨어 큐 스트림(100), DWM 프로세스 스트림(105), 애플리케이션 생성 컨텐츠 스트림(110), 계류중인 플립 스트림(115) 및 온스크린 디스플레이 스트림(120)을 도시한다. 본 도시의 목적상 프레임 렌더링 속도가 디스플레이 리프레시 속도보다 더 빠르다고 가정한다. 다시 본 도시의 목적상 초기에 온스크린

디스플레이 스트림(120)이 도 3에서 단순히 현재 데스크탑 외관(22)이 디스플레이(20) 상에 있는 버퍼1의 컨텐츠를 반영하는 것으로 가정한다. 또한 리프레시1에서 애플리케이션 생성 컨텐츠(A)가 다시 도 2에 도시된 운영체제(80)를 거쳐 생성을 시작하고 DWM 프로세스 웨이크/실행 인스턴스(170)가 발생한다고 가정한다. 또한 리프레시1에서, 그리고 하드웨어 큐 스트림(100)에서 보여지는 바와 같이, 렌더링 하드웨어는 컨텐츠(A)를 렌더링하기 시작한다. 하드웨어 큐 스트림(100)에서 보여지는 바와 같이, 컨텐츠(A)의 렌더링(렌더(A))는 리프레시1 및 리프레시2 사이에서 완료된다. 하지만, 리프레시2가 고정된 시점에서 발생하기 때문에, DWM 합성 프로세스(150)는 리프레시2까지 기다려야 하고, 이는 합성에 있어 지연이 생기게 한다. 다음으로 리프레시2에서, 다음 DWM 프로세스 웨이크/실행 인스턴스(175)가 발생한다. DWM 프로세스 웨이크/실행 인스턴스(175)는 이용가능한 렌더링된 컨텐츠(A)를 감지하고 화살표(177)에서 반영되는 바와 같이 DWM 합성 프로세스(150)를 촉발시킨다. DWM 합성 프로세스(150)는 진행할 수 있고 렌더링된 합성된 데스크탑(22)이 버퍼2에 저장되도록 야기할 수 있다. DWM 합성 프로세스(150)의 끝에서, 버퍼1로부터 버퍼2로의 플립은 발생할 수 없고 리프레시3까지 기다려야 하고 따라서 플립 지연(180)이 생기게 한다. 마지막으로, 리프레시3에서 DWM 프로세스의 다음 인스턴스(185)가 발생하고 버퍼1로부터 버퍼2로의 플립이 온스크린 디스플레이 스트림(120)에서 보여지는 바와 같이 발생한다. 그러므로, 두 개의 풀 프레임들, 즉, 리프레시1로부터 리프레시2까지 그리고 리프레시2로부터 리프레시3까지, 이 (A)의 렌더링 시작으로부터 합성된 데스크탑 외관(22)으로서 (A)의 디스플레이까지 가기 위해 요구된다. 물론 지연은 합성시 지연 및 플립 지연(180)의 결합이다.

[0027] 예시적인 새로운 기술은 이제 도 2, 3 및 6을 참조함으로써 이해될 수 있다. 도 6은 도 5처럼 하드웨어 큐 스트림(100), DWM 프로세스 스트림(105), 애플리케이션 생성 컨텐츠 스트림(110), 계류중인 플립 스트림(115) 및 온스크린 디스플레이 스트림(120)을 도시한다. 도 5에서 도시된 가정처럼, 오직 4개의 리프레시 사이클들, 리프레시1, 리프레시2, 리프레시3 및 리프레시4, 이 도시되고 초기에 버퍼1의 컨텐츠가 도 3에 보여지는 디스플레이(20) 상에 도시되는 것으로 가정한다. 다시 본 비교를 위해, 프레임 렌더링 속도가 디스플레이 리프레시 속도보다 빠르다. 리프레시1 후에, 애플리케이션 생성 컨텐츠(A)는 렌더링(컨텐츠(A)의 렌더(A))를 시작하는 하드웨어 큐 스트림(100)에서 보여지는 바와 같이 렌더링 하드웨어로 전달된다. 도 5에 도시된 종래의 기술처럼 리프레시당 DWM 프로세스 웨이크/실행의 단일 인스턴스를 사용하는 대신, DWM 프로세스 스트림(105)은 그들 중 두개를 190 및 195로 나타낸 다수의 DWM 프로세스 웨이크/실행 인스턴스들이 있는 것을 반영한다. 다수의 DWM 프로세스 웨이크/실행 인스턴스들(190 및 195)은 도 2에서 도시된 고해상도 타이머(92)에 의해 제어된다. 고해상도 타이머(92)로, DWM 합성 프로세스는 일반적으로 디스플레이(20) 상에 부과된 VSYNC 타이밍으로부터 분리된다. 이하에 설명되는 바와 같이, 고해상도 타이머(92)는 다양한 형태의 대기시간 민감 컨텐츠를 감지하는 것에 응답하여 자동으로 인에이블되거나 또는 사용자 입력에 의해 수동으로 인에이블될 수 있다. 더욱이, 리프레시당 DWM 프로세스 웨이크/실행 인스턴스들(190 및 195)의 수는 사용자에 의해 선택되거나, 운영체제에 의해 선택되거나 또는 그와 다를 수 있다. 물론 고해상도 타이머 주파수, 즉 고해상도 타이머(92)가 리프레시들 사이에서 DWM 프로세스 웨이크/실행 인스턴스들을 야기하는 주파수,는 동적이고 고해상도 타이머(92)를 실행하는 것과 연관된 컴퓨팅 오버헤드를 관리하기 위해 컨텐츠에 기초하여 변경될 수 있다. 고해상도 타이머(92)는 운영체제 소프트웨어(80) 또는 비디오 드라이버 소프트웨어(85) 내에, 또는 몇몇 다른 위치에서 구현될 수 있다. 고해상도 타이머(92)는 주기적으로 비디오 드라이버 소프트웨어(85)가 현재 디스플레이를 체크, 즉 어떤 버퍼가, 말하자면 버퍼1 또는 버퍼2, 현재 디스플레이되고 있는지, 하도록 함으로써 주기적으로 DWM 프로세스 인스턴스(190 및 195) 각각을 야기시킨다. 비디오 드라이버 소프트웨어(85)는 그리고는 요청된 체크를 수행하고 운영체제(80)로 체크 결과, 즉 어떤 버퍼, 버퍼1 또는 버퍼2, 가 현재 디스플레이되고 있는지,를 보고한다. 이 보고는 운영체제(80)가 다음 DWM 웨이크/실행 인스턴스를 개시하는 것을 촉발시킨다. DWM 프로세스 웨이크/실행 인스턴스들 각각, 예를 들어 인스턴스(190 및 195),에 대해 DWM 프로세스는 임의의 애플리케이션 할당된 버퍼들, 말하자면 버퍼m,의 컨텐츠들을 조사(poll)하고 렌더링 하드웨어가 DWM 합성 프로세스를 시작해야 하는지에 대한 결정을 한다. 예를 들어, 만일 DWM 프로세스 웨이크/실행 인스턴스(190)가 (A)의 렌더링(렌더(A))이 완료되지 않았을 때 발생하면, 그 시점에서의 DWM 합성은 적절하지 않다. 하지만, DWM 프로세스 웨이크/실행 인스턴스(195)시, (A)의 렌더링(렌더(A))가 완료되고 그 렌더링된 컨텐츠(A)는 애플리케이션 할당된 버퍼인 버퍼m에 저장된다. 그 시점에 DWM 프로세스 웨이크/실행 인스턴스(195)는 렌더링 하드웨어가 화살표(203)에 의해 제안되는 바와 같이 그리고 하드웨어 큐 스트림(100)에서 반영되는 바와 같이 DWM 합성 프로세스(200)를 시작하도록 한다. 이것은 여기에 개시된 종래 기술들보다 더 빠른, 리프레시2 이전에 발생한다는 것이 주목된다. DWM 합성 프로세스(200) 동안, 데스크탑 합성자는 애플리케이션 할당된 버퍼인 버퍼m의 컨텐츠 및 데스크탑 외관(22)의 임의의 다른 부분을 포함하는 합성된 외관(22)을 합성하고, 그 합성된 외관(22)을 버퍼2에 저장한다. 이 시점에서, 계류중인 플립 스트림(115)에서 보여지는 바와 같이 버퍼1로부터 버퍼2로의 플립은 계류중이나 리프레시2까지 발생하지 않고, 플

립 지연(205)을 야기한다. 하지만, 리프레시1 및 리프레시2 사이에서 DWM 프로세스 웨이크/실행의 다수의 인스턴스들(190, 195, 등)이 있으므로, 도 5에서 보여지는 종래 기술에서보다 DWM 합성 단계(200)의 매우 더 빠른 실행이 발생할 수 있고, 상대적으로 짧은 플립 지연(205) 및 궁극적으로 리프레시2에서 발생할 수 있는 버퍼2로의 플립을 야기한다. 그러므로, 렌더(A)의 시작으로부터 DWM 합성(200)으로 버퍼2로의 플립으로의 프로세스 그리고 버퍼2의 컨텐츠의 온스크린 디스플레이는 도 5에서 도시된 종래의 기술에 대한 두 개의 프레임들과는 대조적으로 단일 프레임에서 발생할 수 있다. FreeSync가 개시된 실시예들과 같이 사용될 수 있다는 점이 주목된다.

[0028] 프레임들을 렌더링하기 위한 종래 기술 및 새로운 대안적인 예시적 기술 사이의 유사한 비교, 하지만 이와 관련하여 초당 렌더링 프레임 속도가 모니터의 리프레시 속도보다 더 낮은, 가 이제 도 2, 3, 7 및 8을 참조하여 이해될 수 있다. 도 7은 도 5와 같은 활동도이고 도 8은 도 6과 같은 활동도이다. 여기에서, 도 7은 다시 하드웨어 큐 스트림(100), DWM 프로세스 스트림(105), 애플리케이션 생성 컨텐츠 스트림(110), 계류중인 플립 스트림(115), 온스크린 디스플레이 스트림(120) 및 4개의 리프레시 사이클들, 리프레시1, 리프레시2, 리프레시3 그리고 리프레시4,을 도시한다. 다시, 초기에 디스플레이(20) 상의 데스크탑 외관(22)은 도 2에서 보여지는 버퍼1의 컨텐츠임을 가정한다. 리프레시1 바로 후에, 애플리케이션 생성 컨텐츠 스트림(110) 및 하드웨어 큐 스트림(100)은 컨텐츠(A)가 렌더링 하드웨어로 전달되고 (A)의 렌더링(렌더(A))이 시작하는 것을 반영한다. DWM 프로세스 웨이크/실행 인스턴스(205)는 발생한다. 리프레시2에서, 다음 DWM 프로세스 웨이크/실행 인스턴스(210)가 발생하나 렌더링 프레임 속도가 리프레시 속도, 즉 리프레시1로부터 리프레시2까지의 기간,보다 더 늦으므로, 하드웨어 큐 스트림(100)에서 반영되는 바와 같이 컨텐츠(A)의 렌더링(렌더(A))이 완료되지 않는다. 비록 렌더(A)가 리프레시2에 이어 그리고 리프레시3 전에 완료된다 할 지라도, 이 완료는 DWM 프로세스 웨이크/실행 인스턴스(210)의 끝 이후 발생한다. 그러므로 렌더링된 컨텐츠, 렌더(A),의 DWM 합성으로의 계속된 이동 또는 그 반대는 적어도 리프레시3까지 지연된다. 리프레시3에서, 다음 DWM 프로세스 웨이크/실행 인스턴스(215)가 발생하고, 렌더(A) 프로세스로부터 렌더링된 컨텐츠가 감지되고 렌더링 하드웨어는 화살표(223)에 의해 제안되는 바와 같이 DWM 합성 프로세스(220)를 시작하도록 한다. DWM 합성 프로세스(220) 동안, 렌더링된 컨텐츠(A) 및 데스크탑 외관의 임의의 남아있는 부분들로 구성되는 합성된 데스크탑(22)은 버퍼2에 기입된다. DWM 합성 단계(220)의 완료에 이어, 합성된 데스크탑 외관(22)을 디스플레이(20)로 실제로 전달하기 위한 버퍼2로의 플립은 리프레시4까지 기다려야 하고, 이는 커다란 플립 지연(225)을 생기게 한다. 마지막으로 리프레시4에서, 버퍼2로의 플립은 온스크린 스트림(120) 상에 나타내지는 바와 같이 업데이트된 데스크탑 외관(22)(업데이트된 컨텐츠(A)를 포함하여)을 모니터(20)로 전달하기 위해 발생할 수 있다. 하드웨어 큐(100)에서 렌더(A)의 완료에 뒤이은 것과 연관된 전체 지연은 1 프레임이 넘고 2 프레임만큼 클 수 있다. 더욱이, 3개의 풀 프레임들, 리프레시1에서 리프레시4까지, 이 렌더(A)의 시작 및 데스크탑 외관(22)의 일부로서 (A)의 실제 디스플레이 사이에서 발생하였다.

[0029] 대조적으로, 그리고 도 8에서 보여지는 바와 같이, 다수의 DWM 프로세스 웨이크/실행 인스턴스들을 생성하기 위해 고해상도 타이머(92)(도 2 참조)를 이용하는 새로운 예시적인 기술은 도 8에 보여진다. 다시, 도 8은 하드웨어 큐 스트림(100), DWM 프로세스 스트림(105), 애플리케이션 생성 컨텐츠 스트림(110), 계류중인 플립 스트림(115) 및 온스크린 디스플레이 스트림(120)을 도시한다. 하지만 도 3에 도시된 모니터(20)에 대한 동적 리프레시 속도의 이용은 리프레시들, 리프레시1, 리프레시2, 리프레시3 및 리프레시4,이 고정된 간격이 아닌 오히려 DWM 합성 프로세스의 시점인 임의의 동적 리프레시 속도에 의해 결정되는 간격으로 발생한다는 것을 의미한다. 다시, 초기에 버퍼1의 컨텐츠가 디스플레이된다. 리프레시1에 이어, 애플리케이션 생성 컨텐츠(A)가 도 2에 도시된 운영체제(80)에 의해 호출되고 그러므로 하드웨어 큐(100)는 렌더(A)를 시작하는 것을 또한 보여준다. 도 2에서 도시된 고해상도 타이머(92)는 그들 중 둘을 230 및 235로 각각 나타낸 다수의 DWM 프로세스 웨이크/실행 인스턴스들을 생성한다. 종래의 맥락에서, 리프레시2는 수직의 점선으로 도시된 종래의 VSYNC 리프레시 시점이 발생하는 경우 발생할 것이라는 점이 주목된다. 하지만, 동적 리프레시가 구현되기 때문에, 리프레시2는 나중에 그러므로 VSYNC에 의해 제공되는 것보다는 리프레시1 및 리프레시2 사이에서 더 낮은 리프레시 속도로 일어날 수 있다. 그러므로, 몇몇 개수의 DWM 프로세스 웨이크/실행 인스턴스들, 그리고 이 경우에는 DWM 프로세스 웨이크/실행 인스턴스(235), 이후 (A)의 렌더링(렌더(A))이 완료되고 이것이 DWM 프로세스 웨이크/실행 인스턴스(235)에 의해 감지된다. DWM 프로세스 웨이크/실행 인스턴스(235)는 화살표(243)에 의해 제안되고 하드웨어 큐 스트림(100)에 반영된 바와 같이 렌더링 하드웨어가 DWM 합성 프로세스(240)를 시작하도록 신호를 보낸다. DWM 합성 프로세스(240) 동안, 렌더링 하드웨어는 렌더링된 컨텐츠(A) 및 데스크탑 외관(22)의 임의의 다른 부분(도 3 참조)을 포함하는 합성된 외관을 렌더링하고 그 합성된 데스크탑 외관(22)을 버퍼2에 기입한다. DWM 합성 프로세스(240)의 끝에서, 사용자 구성가능하고 잠재적으로 매우 짧은 플립 지연(245)이 리프레시2 및 버퍼1로부터 버퍼2로의 플립에 이어 즉시 발생할 수 있다. 플립 지연(245)의 길이는 고해상도 타이머(92)에 의해 설정될 수 있고 다양한 가치들을 지닐 수 있다. 예시적인 실시예에서, 예시들은 0.5에서 20 밀리초(millisecond)를 포함한

다. 그러므로, 합성된 외관(22)(컨텐츠 (A) 포함)을 전달하기 위해 3개의 풀 프레임을 필요로 하는 대신에, 동적 리프레시 및 더 빈번한 DWM 프로세스 웨이크/실행 인스턴스들 단계들(230 및 235 등)를 이용하는 새로운 기술은 컨텐츠(A)가 단일의 동적으로 리프레시된 프레임으로 전달될 수 있다. 고해상도 타이머(92)의 주파수는 정적 또는 동적일 수 있다는 것이 주목된다. 예를 들어, 다수의 DWM 프로세스 웨이크/실행 인스턴스들을 실행하는 것과 연관된 컴퓨팅 오버헤드를 줄이기 위해, 고해상도 타이머 주파수가 컨텐츠에 따라 동적으로 조정될 수 있다. 예를 들어, 게임 애플리케이션은 높은 프레임 속도를 생성할 수 있다. 이러한 높은 프레임 속도 컨텐츠의 렌더링 동안, 고해상도 타이머(92)의 주파수는 동적으로 특정 주파수 ω_1로 설정될 수 있다. 하지만, 게임 애플리케이션이 더 낮은 프레임 속도를 갖는 주기들이 있을 수 있다. 일 예로는 애플리케이션이 시작한 후 인트로 스크린들이 디스플레이되는 처음 몇초간일 수 있다. 이러한 낮은 프레임 속도의 계산 강도(computational intensity) 주기 동안, 고해상도 타이머(92)는 또 다른 주파수 ω_2, 여기서 $\omega_2<\omega_1$인, 로 바뀔 수 있다. 이러한 동적 주파수 스위칭은 초당 여러번 그리고 컨텐츠에 따라 다수의 수파수로 발생할 수 있다.

[0030] 여기에서 설명된 종래의 렌더링 기술에 비해 개시된 예시적인 실시예들과 연관된 다른 장점은 이제 도 9 및 10을 참조함으로써 이해될 수 있다. 본 도시된 실시예에서, 개선된 렌더링 프로세스는 도 2에 도시된 데스크탑 합성자(90)에게 플립이 발생하였으므로 두 버퍼들 중 하나는 해방되어 새로운 컨텐츠의 렌더링을 시작할 수 있다는 것을 알리는 시점을 시간적으로 전진시킨다. 도 9는 도 5와 같은 활동도이고 종래의 렌더링 프로세스를 도시한다. 다시, 도 9는 하드웨어 큐 스트림(100), DWM 프로세스 스트림(105), 애플리케이션 생성 컨텐츠 스트림(110), 계류중인 플립 스트림(115) 및 온스크린 디스플레이 스트림(120) 그리고 4개의 리프레시 사이클들, 리프레시1, 리프레시2, 리프레시3 및 리프레시4,을 도시한다. 리프레시1에서, 제1 DWM 프로세스 웨이크/실행 인스턴스(250)가 발생하고 애플리케이션 생성 컨텐츠(A)가 하드웨어 큐 스트림(100)에서 반영되는 바와 같이 업데이트된 컨텐츠(A)를 애플리케이션 할당된 버퍼m에 렌더링하는 프로세스가 시작하는 렌더링 하드웨어로 운영체제에 의해 보내진다. 이 시점에서 디스플레이(20)는 다시 데스크탑 합성자(90) 할당된 버퍼들 중의 하나인 버퍼 1의 컨텐츠를 보여주고 있다는 것을 가정한다. 컨텐츠(A)의 버퍼m에의 렌더링은 리프레시2 이전에 완료된다. 컨텐츠(A)의 버퍼m에의 렌더링 끝에, 애플리케이션 생성 컨텐츠(B)가 또다른 애플리케이션 할당된 버퍼인 버퍼n에의 렌더링을 위해 선발된다(slated). 하지만 이하에 더욱 상세히 설명되는 바와 같이 이것은 나중까지 발생할 수 없다. 다음으로, 리프레시2에서, 다음 DWM 프로세스 웨이크/실행 인스턴스(255)가 발생한다. DWM 프로세스 웨이크/실행 인스턴스(255)는 버퍼m에서 이용 가능한 렌더링된 컨텐츠(A)를 감지하고 렌더링 하드웨어가 화살표(262)에 의해 제안되고 하드웨어 큐 스트림(100)에 반영된 바와 같이 DWM 합성 프로세스(260)를 시작하도록 한다. DWM 합성 프로세스(260)는 렌더링된 컨텐츠(A)를 포함하는 합성된 외관(22)(도 3 참조)을 생성한다. DWM 합성 프로세스(260)의 끝에 이어, 버퍼1로부터 버퍼2로의 플립이 다음 리프레시인 리프레시3을 기다려야 하고, 이는 플립 계류중인 스트림(115)에서 보여지는 플립 지연(265)을 생기게 한다. 리프레시3에서, 버퍼2로의 플립이 발생하여 버퍼2의 컨텐츠, 즉 렌더(A)의 결과물, 가 도 3에 보여지고 온스크린 디스플레이 스트림(120)에 반영되는 바와 같이 디스플레이(20)로 전달된다. 종래 기술에서의 이 시점에서, DWM 프로세스 인스턴스(270)는 디스플레이되고 있는 현재 버퍼를 보고하고 이 경우에는 버퍼2가 될 것이다. 오직 이 시점에서만 운영체제(80)가 이 현재 버퍼 디스플레이의 보고를 수신하고 다른 버퍼, 버퍼1, 가 다음 합성된 컨텐츠, 이 경우에는 컨텐츠(B)를 포함하는,를 렌더링하기 위해 이제 이용가능하다는 것을 알게 된다.

[0031] 새로운 프로세스의 예시적인 실시예는 도 10에 도시되고 도 2 및 3과 함께 설명될 것이다. 도 9와 같이, 도 10은 하드웨어 큐 스트림(100), DWM 프로세스 스트림(105), 애플리케이션 생성 컨텐츠 스트림(110), 계류중인 플립 스트림(115) 및 온스크린 디스플레이 스트림(120) 그리고 4개의 리프레시 사이클들, 리프레시1, 리프레시2, 리프레시3 및 리프레시4,를 도시한다. 리프레시1에서, 버퍼1의 컨텐츠가 디스플레이되고 애플리케이션 생성 컨텐츠(A)가 하드웨어 큐 스트림(100)을 통해서 반영되는 바와 같이 렌더(A)를 애플리케이션 할당된 버퍼m에 하는 프로세스로 시작하기 위해 렌더링 하드웨어로 보내진다. 도 2에 도시된 고해상도 타이머(92)를 거쳐, 다수의 주기적인 DWM 프로세스 웨이크/실행 인스턴스들(275, 280, 등)이 리프레시1 이후 리프레시2 전에 발생한다. 위에서 논의된 대로, 고해상도 타이머(92)는 비디오 드라이버 소프트웨어(85)가 현재 디스플레이, 즉 어떠한 DWM 할당된 버퍼, 말하자면 버퍼1 또는 버퍼2, 가 현재 디스플레이되고 있는지, 인지를 주기적으로 체크하도록 한다. 비디오 드라이버 소프트웨어(85)는 요청된 체크를 수행하고 운영체제(80)로 체크 결과, 즉 어떤 버퍼, 버퍼1 또는 버퍼2,가 현재 디스플레이되고 있는지를 보고한다. 이 보고는 운영체제(80)가 다음 DWM 웨이크/실행 인스턴스를 개시하는 것을 촉발시킨다. 버퍼m에의 렌더(A) 프로세스의 결과는 다음 이용가능한 DWM 프로세스 웨이크/실행 인스턴스, 이 경우는 인스턴스(280),에 의해 감지된다. DWM 프로세스 웨이크/실행 인스턴스(280)는 렌더링 하드웨어에게 화살표(287)에 의해 제안되는 바와 같이 DWM 합성 프로세스(285)를 시작하도록 신호를 보낸다.

DWM 합성 프로세스 동안, 버퍼m의 컨텐츠는 데스크탑 외관(22)의 임의의 남아있는 부분과 합성되고(도 3 참조) 버퍼2에 기입된다. 추가적으로, 그 시점에서, 고해상도 타이머(92)에 의해 촉발되는 바와 같이 비디오 드라이버 소프트웨어(85)는 운영체제 소프트웨어(80)로 현재 디스플레이되는 버퍼가 여전히 버퍼1이라고 보고한다. 그러므로, 각각의 DWM 프로세스 웨이크/실행 인스턴스들(275, 280, 등)의 시작시에, 보고가 (현재 DWM 프로세스 웨이크/실행 인스턴스에 의해) 현재 디스플레이되는 버퍼는 버퍼1이라는 것 및 DWM 합성 단계(285) 이후 현재 디스플레이되는 버퍼의 보고를 포함하여 도 2에 도시된 운영체제 소프트웨어(80)의 데스크탑 합성자(90)로 보내진다. 동일한 보고가 리프레시2 전 DWM 웨이크/실행의 매 인스턴스(275, 280, 등)시에 반복된다. 리프레시2에서, 버퍼1로부터 버퍼2로의 플립이 행해지고 모니터(20)로 전달된다. 리프레시2 후 즉시 발생하는 다음 DWM 프로세스 웨이크/실행 인스턴스(290)시, 보고는 현재 디스플레이되는 버퍼는 버퍼2, 그러므로 버퍼1이 새로운 합성을 위해 이용가능하다는 것을 나타낸다. 본 예시기술에서 다음 이용가능한 버퍼의 이러한 보고는 리프레시3에서 발생하는 도 9에서 도시된 현재 디스플레이되는 버퍼의 보고보다 매우 빠르게 발생한다. 다시 이것은 하드웨어 지연에 따른 대기시간을 줄이는데 도움이 된다.

[0032] 컴퓨팅 디바이스(10)의 동작에 대한 예시적인 프로세스 흐름은 이제 도 2, 3, 8 및 11을 참조하여 이해될 수 있다. 도 11은 예시적인 프로세스 흐름의 흐름도이다. 단계(300)에서 시작한 후, 컴퓨팅 디바이스(10)는 만일 전체화면 전용 모드가 인에이블되었는지를 단계(305)에서 감지한다. 만일 전체 화면 전용 모드가 감지되면, 프로세스는 단계(310)로 진행하여 고해상도 타이머를 디스에이블한 채로 전체화면 전용 모드로 렌더링한다. 프로세스는 단계(305)로 되돌아간다. 만일, 하지만, 단계(305)에서 전체화면 전용 모드가 감지되지 않으면, 프로세스는 단계(320)로 가고 대기시간 민감 컨텐츠를 감지한다. 이것은 다양한 방식으로 행해질 수 있다. 예시적인 실시예에서, 도 2에서 도시된 프로세서(50)는 비디오 드라이버 소프트웨어(85)와 함께 애플리케이션(35), 게임 또는 다른 것과 같은,의 전체화면 모드 또는 윈도우 모드로의 개시(launch)를 감지한다. 이것은 윈도우가 전체화면인지 아닌지를 결정하기 위해 윈도우 사이즈 및 전경 및 배경 캐릭터가 감지되는지 그리고 데스크탑 사이즈가 비교되는 실행 시간(run time) 감지를 수반할 수 있다. 이 감지 단계는 고해상도 타이머(92)와 연관된 임의의 추가적인 오버헤드 및 보다 빈번한 데스크탑 DWM 프로세스 웨이크 및 인에이블 단계들이 방지되도록 대기시간 민감 애플리케이션들이 실행되고 있는 환경에서만 고해상도 타이머(92)를 인에이블하는 것이 유리하다. 만일 단계(323)에서 대기시간 민감 컨텐츠가 감지되지 않으면 프로세스는 단계(325)로 진행하여 고해상도 타이머(92)가 디스에이블된 채로 윈도우 모드의 렌더링을 한다. 이 윈도우 모드는 애플리케이션이 전체 화면으로 렌더링되는 모드를 포함할 수 있다. 다음으로 단계(330)에서, 프로세스는 단계(305)로 돌아간다. 만일, 하지만, 단계(323)에서 대기시간 민감 컨텐츠가 감지되면, 프로세스는 고해상도 타이머(92)가 인에이블된 채로 컨텐츠가 렌더링되는 단계(340)로 움직인다. 그러므로 예를 들어, 도 8에 도시된 바와 같이 이것은 다수의 DWM 프로세스 웨이크/실행 인스턴스들(230, 235, 등)을 유발하는 고해상도 타이머(92)를 수반할 수 있다. 다음으로 단계(345)에서, DWM 합성 프로세스가 수행된다. 이것은 예를 들어 도 8에 도시된 DWM 합성 프로세스(240)에 해당한다. 다음으로 단계(350)에서 동적 리프레시 및 버퍼 플립이 실행된다. 이것은 예를 들어 동적 리프레시, 리프레시2, 및 버퍼1로부터 버퍼2로의 버퍼 플립에 해당한다. 다음으로 단계(355)에서 도 2에 보여지는 고해상도 타이머(92)는 도 2에 도시된 데스크탑 합성자(90)에게 두 개의 버퍼들 중 하나가 현재 이용가능하다고 보고한다. 이것은 예를 들어 도 10에 도시된 리프레시2에서 현재 디스플레이되는 버퍼 리포트 리포트에 해당한다. 이 시점에서, 프로세스는 단계(330)로 진행하고 궁극적으로는 단계(305)로 돌아갈 수 있다. 물론, 이 루프(loop)는 초당 여러번 발생할 수 있다.

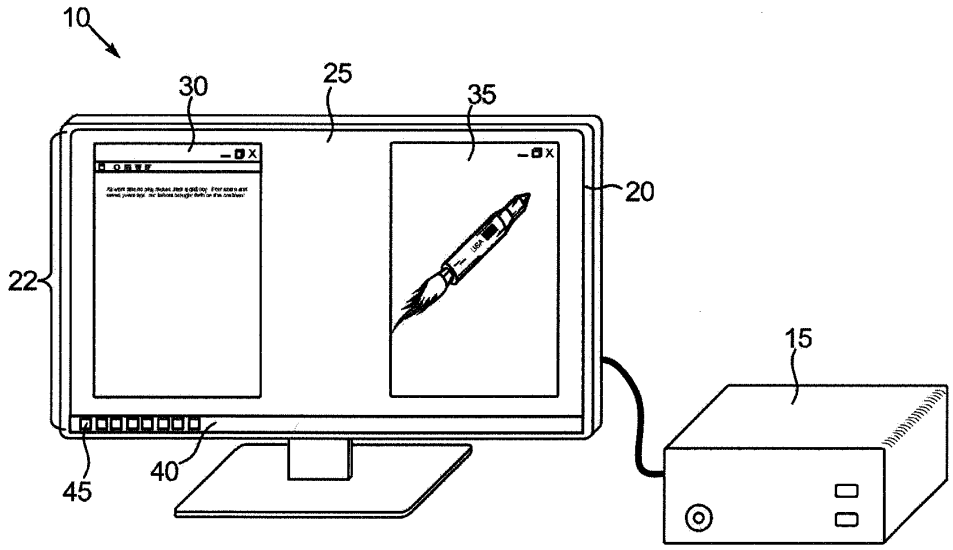
[0033] 도 11의 단계(340)의 몇몇 추가적인 예시적인 상세 내용은 이제 또한 도 12, 이 또한 흐름도인,를 참조함으로써 이해될 수 있다. 도 8의 논의에서 상기 주목된 바와 같이, 고해상도 타이머(92)의 주파수는 정적 또는 동적일 수 있다. 그러므로 단계(340) 내에서, 만일 단계(360)에서 고해상도 타이머 동적 주파수가 인에이블되지 않으면, 단계(365)에서 컨텐츠는 정적 주파수로 고해상도 타이머(92)로 렌더링되고 이어 루프에 의해 단계(360)로 되돌아간다. 고해상도 타이머 주파수는 사용자 입력에 의해 또는 대기시간 민감 컨텐츠가 감지되면 자동으로 설정될 수 있다. 하지만, 만일 단계(360)에서 고해상도 타이머 동적 주파수가 인에이블되면, 단계(370)에서 컨텐츠는 컨텐츠에 기반한 동적 주파수로 고해상도 타이머(92)로 렌더링되고 이어 루프에 의해 단계(360)로 되돌아간다. 주파수는 만일 바람직하다면 컨텐츠의 계산 집약적인(computationally intensive) 측면들에 기반하여 초당 여러번 위아래로 조정될 수 있다.

[0034] 본 발명은 다양한 수정들 및 대안적인 형태들이 가능할 수 있지만, 특정 실시예들이 도면들의 예에 의해 보여졌고 여기에 상세히 설명되었다. 하지만, 본 발명은 개시된 특정 형태들에 한정되도록 의도되지 않았음은 이해되어야 한다. 오히려, 본 발명은 다음에 덧붙인 청구항들에 의해 정의되는 바와 같이 본 발명의 정신 및 범위 내

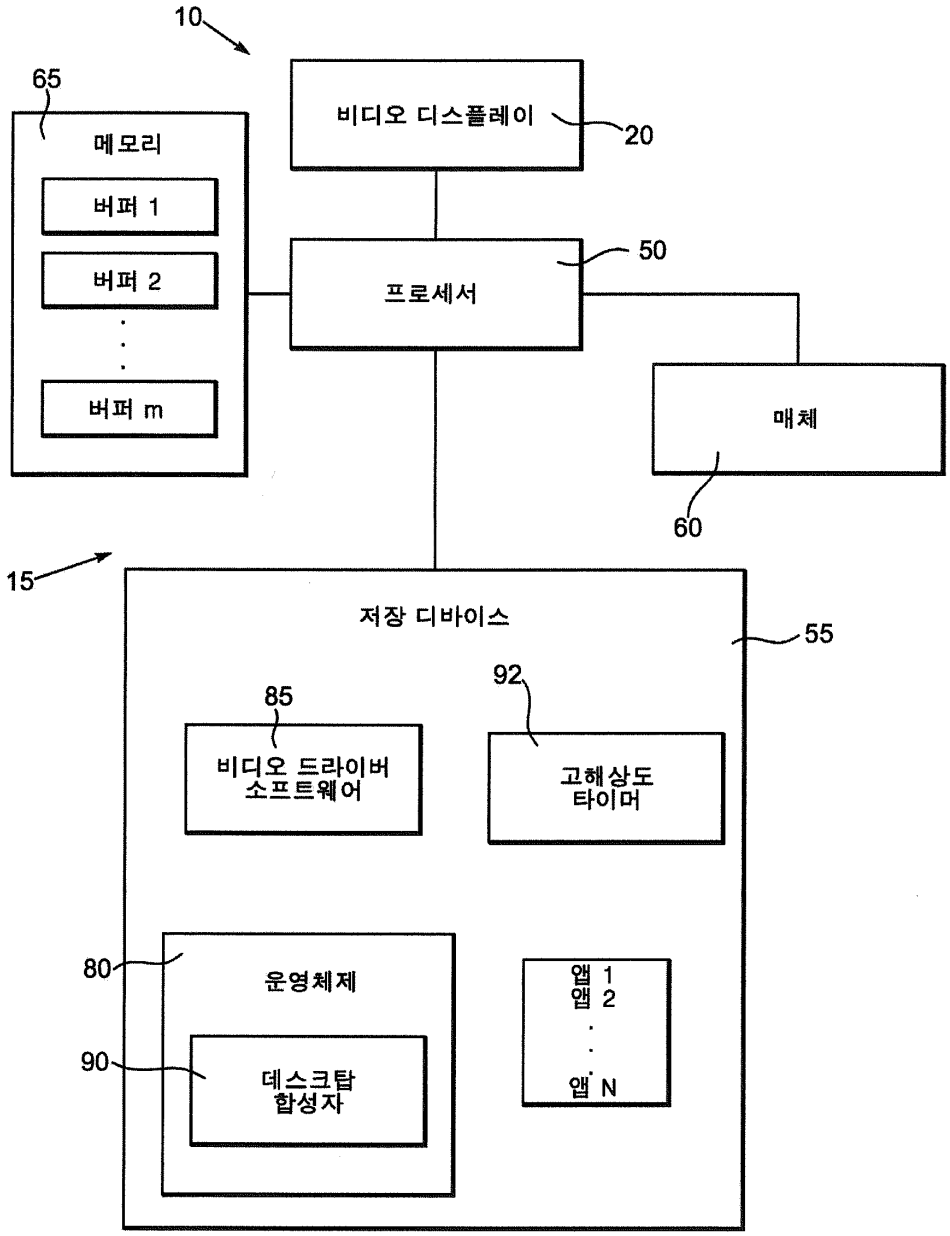
에 속하는 모든 변형들, 동등물들 및 대안들을 포함하는 것이다.

도면

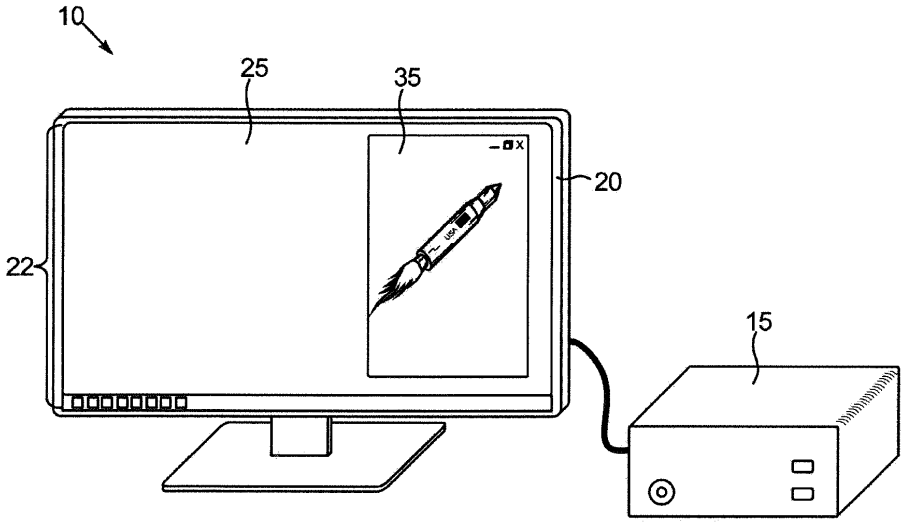
도면1



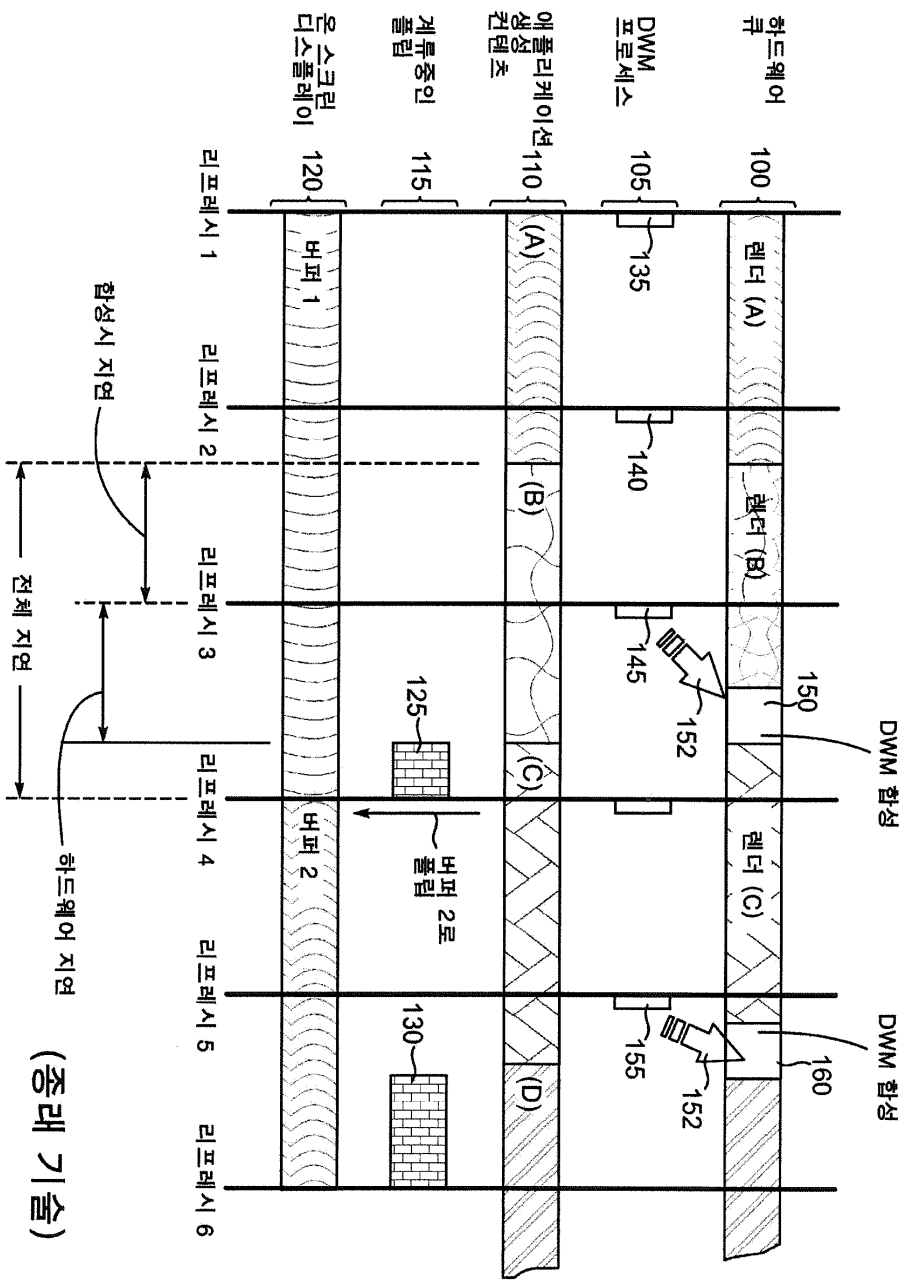
도면2


10
65
메모리
버퍼 1
버퍼 2
버퍼 m
비디오 디스플레이
20
프로세서
50
매체
60
15
저장 디바이스
55
85
비디오 드라이버 소프트웨어
92
고해상도 타이머
80
운영체제
90
데스크탑 합성자
앱 1
앱 2
앱 N

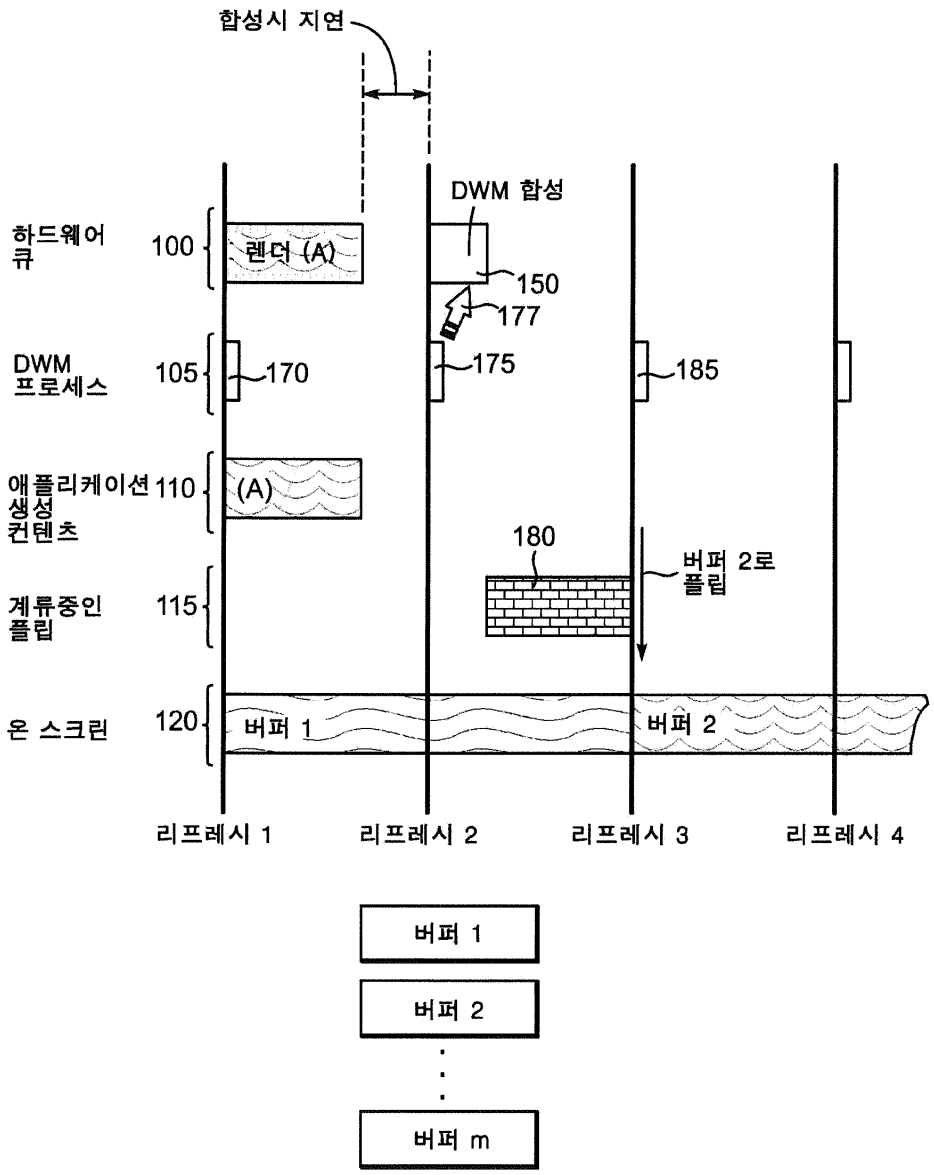
도면3



도면4

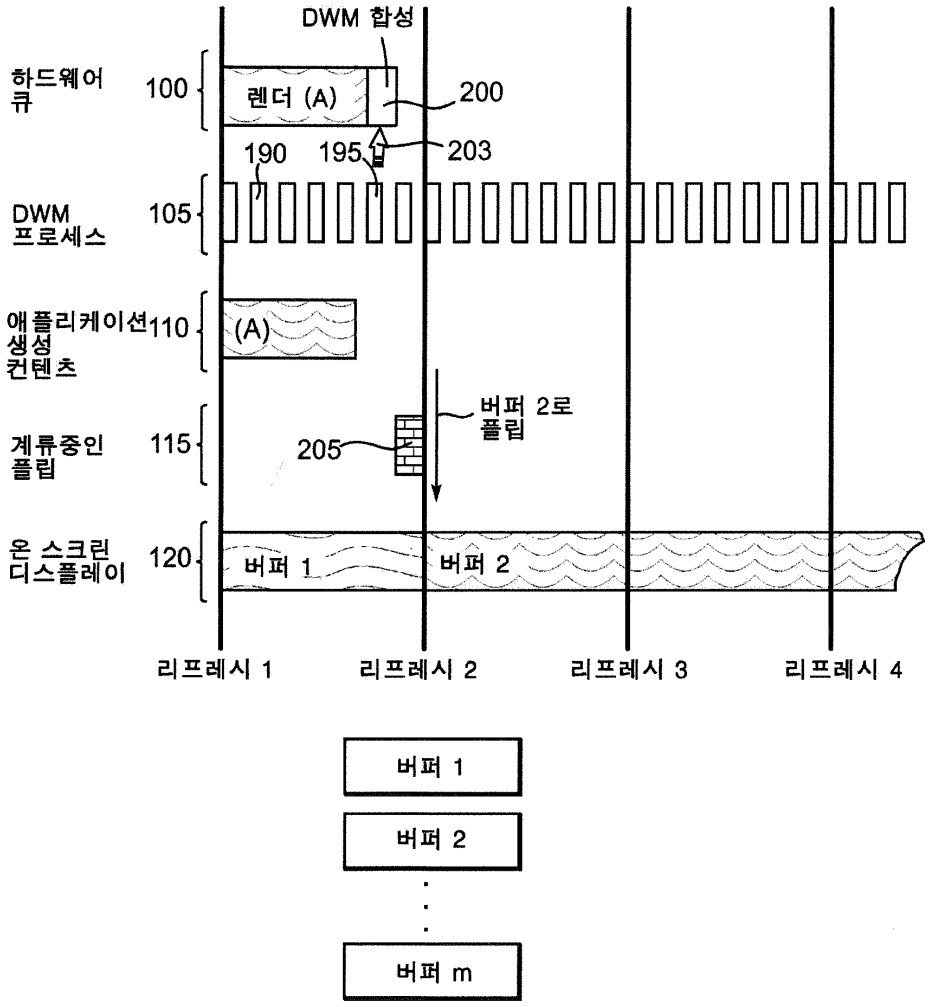

하드웨어 큐
100
DWM 프로세스
105
애플리케이션 생성 컨텐츠
110
계류중인 플립
115
온 스크린 디스플레이
120
렌더 (A)
렌더 (B)
렌더 (C)
버퍼 1
버퍼 2
(A)
(B)
(C)
(D)
135
140
145
150
152
155
160
125
130
DWM 합성
DWM 합성
버퍼 2로 플립
리프레시 1
리프레시 2
리프레시 3
리프레시 4
리프레시 5
리프레시 6
합성시 지연
전체 지연
하드웨어 지연
(종래 기술)

도면5

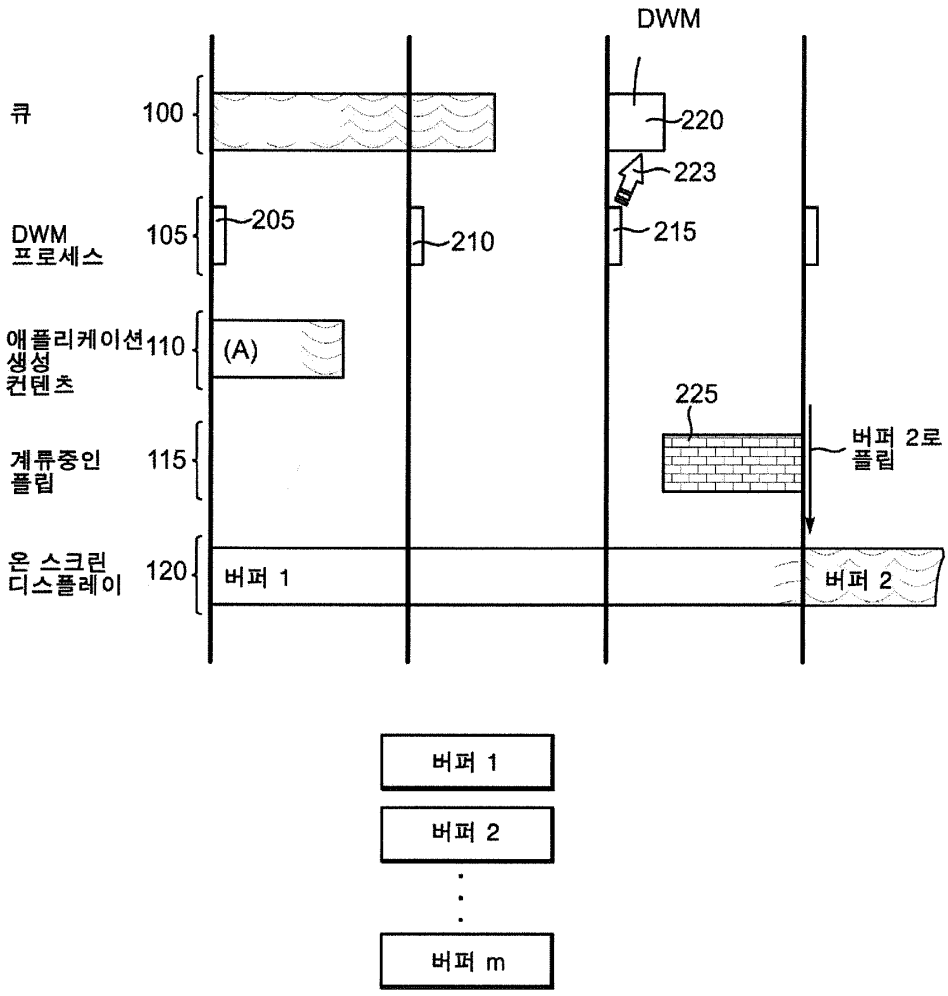

합성시 지연
DWM 합성
하드웨어 큐
100
렌더 (A)
150
177
DWM 프로세스
105
170
175
185
애플리케이션 생성 컨텐츠
110
(A)
180
버퍼 2로 플립
계류중인 플립
115
온 스크린
120
버퍼 1
버퍼 2
리프레시 1
리프레시 2
리프레시 3
리프레시 4
버퍼 1
버퍼 2
버퍼 m

(종래 기술)

도면6

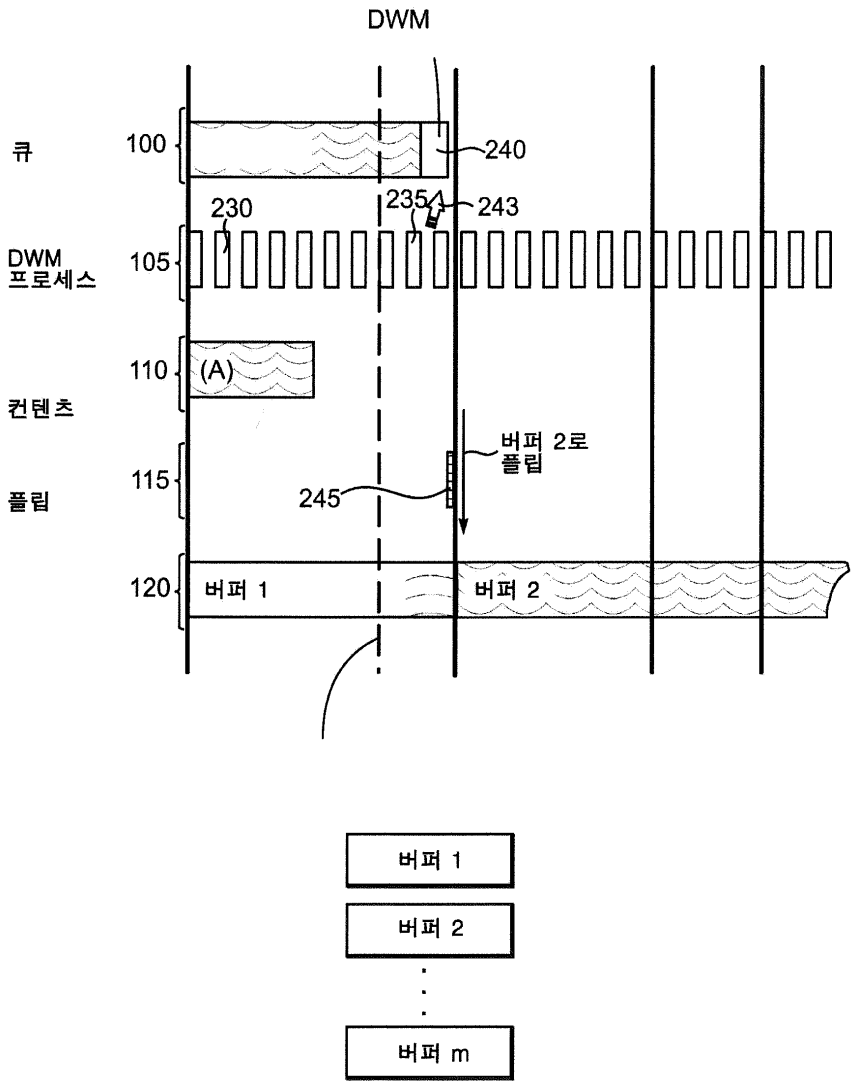

DWM 합성
하드웨어 큐
100
렌더 (A)
200
190
195
203
DWM 프로세스
105
애플리케이션 생성 컨텐츠
110
(A)
버퍼 2로 플립
205
계류중인 플립
115
온 스크린 디스플레이
120
버퍼 1
버퍼 2
리프레시 1
리프레시 2
리프레시 3
리프레시 4
버퍼 1
버퍼 2
버퍼 m

도면7


DWM
큐
100
220
223
DWM
프로세스
105
205
210
215
애플리케이션
생성
컨텐츠
110
(A)
225
버퍼 2로
플립
계류중인
플립
115
온 스크린
디스플레이
120
버퍼 1
버퍼 2
버퍼 1
버퍼 2
버퍼 m

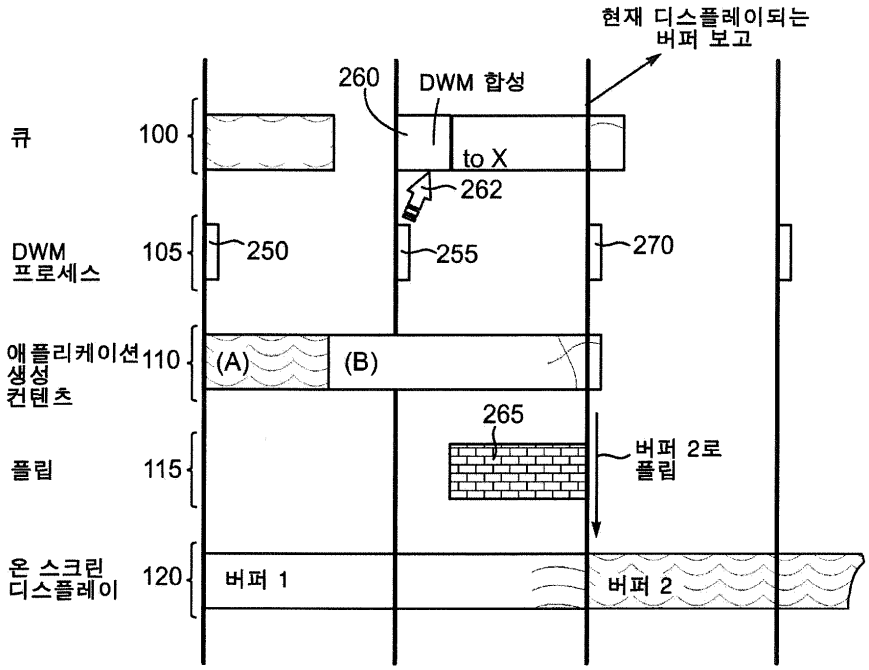
(종래 기술)

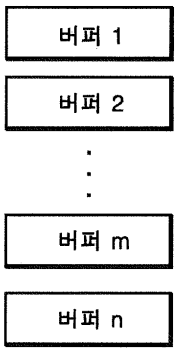
도면8


DWM
큐
100
240
230
235
243
DWM
프로세스
105
110
(A)
컨텐츠
버퍼 2로
플립
115
245
플립
120
버퍼 1
버퍼 2

버퍼 1
버퍼 2
버퍼 m

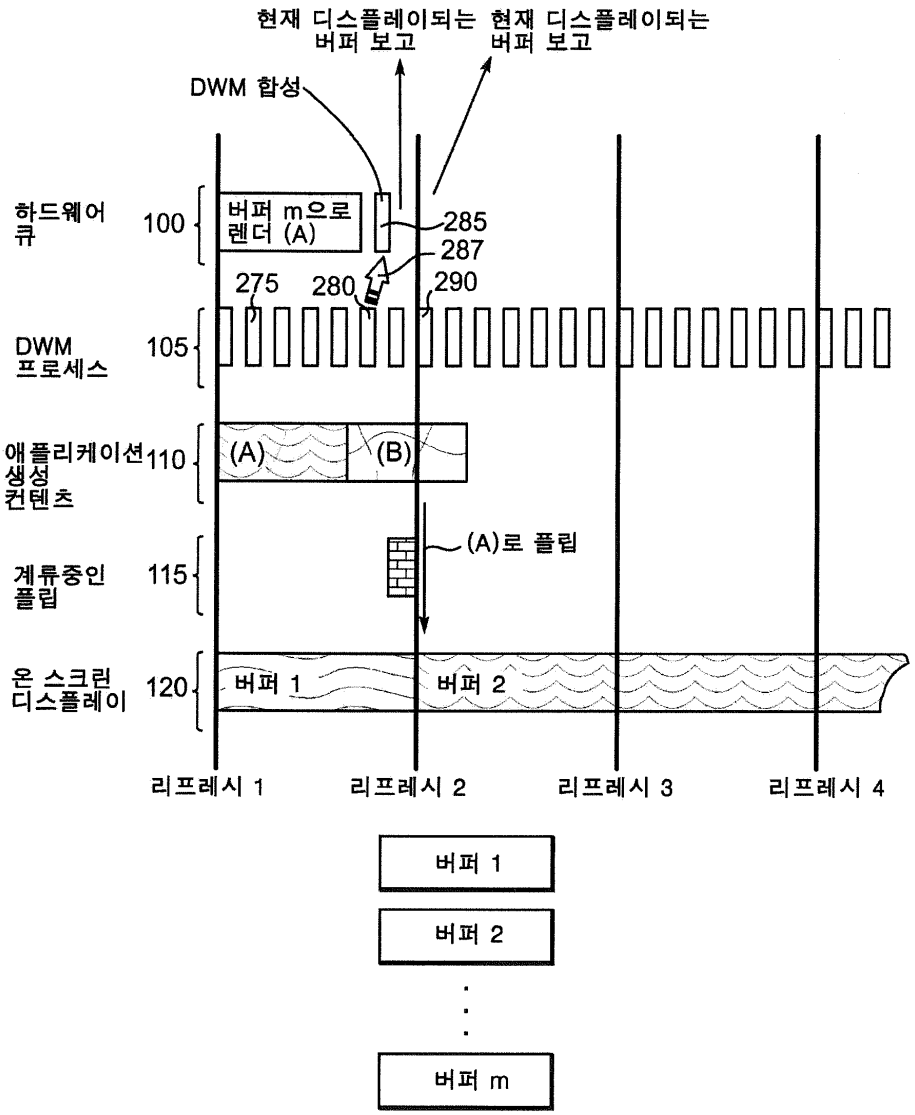
도면9


현재 디스플레이되는
버퍼 보고
260
DWM 합성
to X
262
큐
100
DWM
프로세스
105
250
255
270
애플리케이션
생성
컨텐츠
110
(A)
(B)
265
플립
115
버퍼 2로
플립
온 스크린
디스플레이
120
버퍼 1
버퍼 2

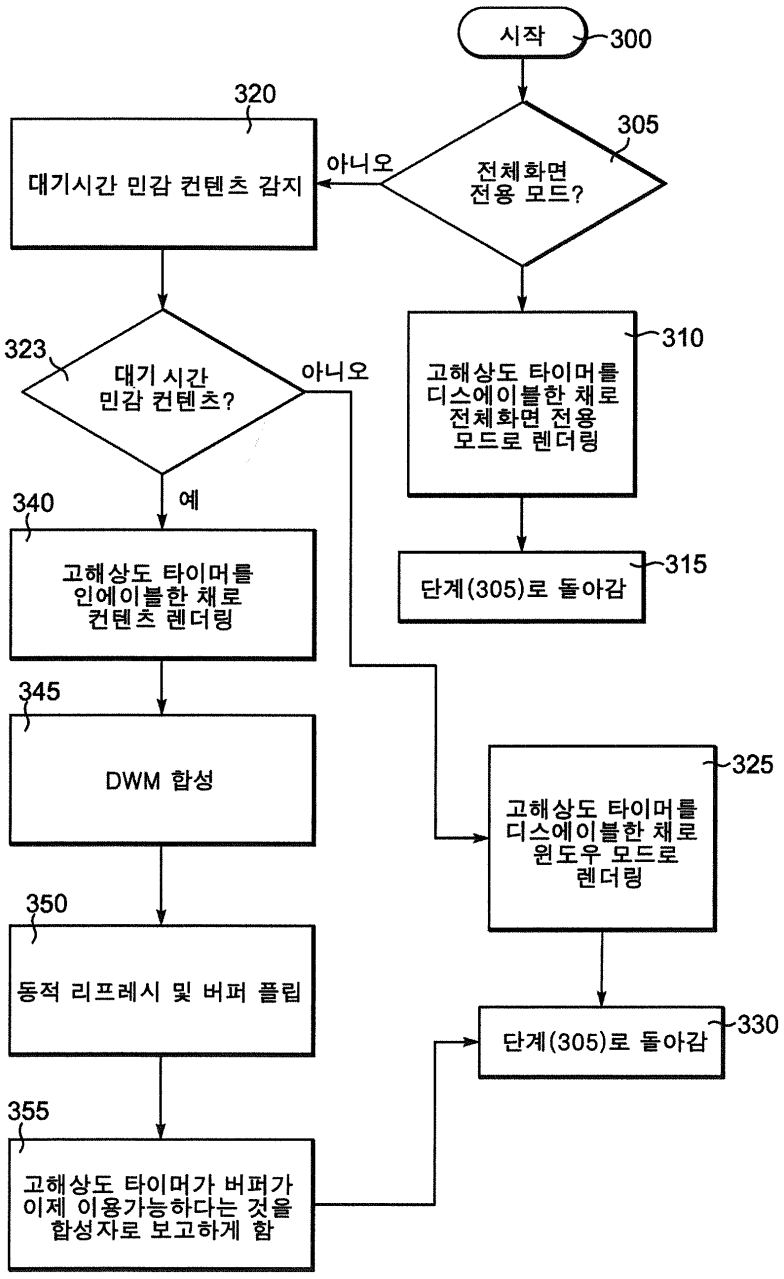

버퍼 1
버퍼 2
버퍼 m
버퍼 n

(종래 기술)

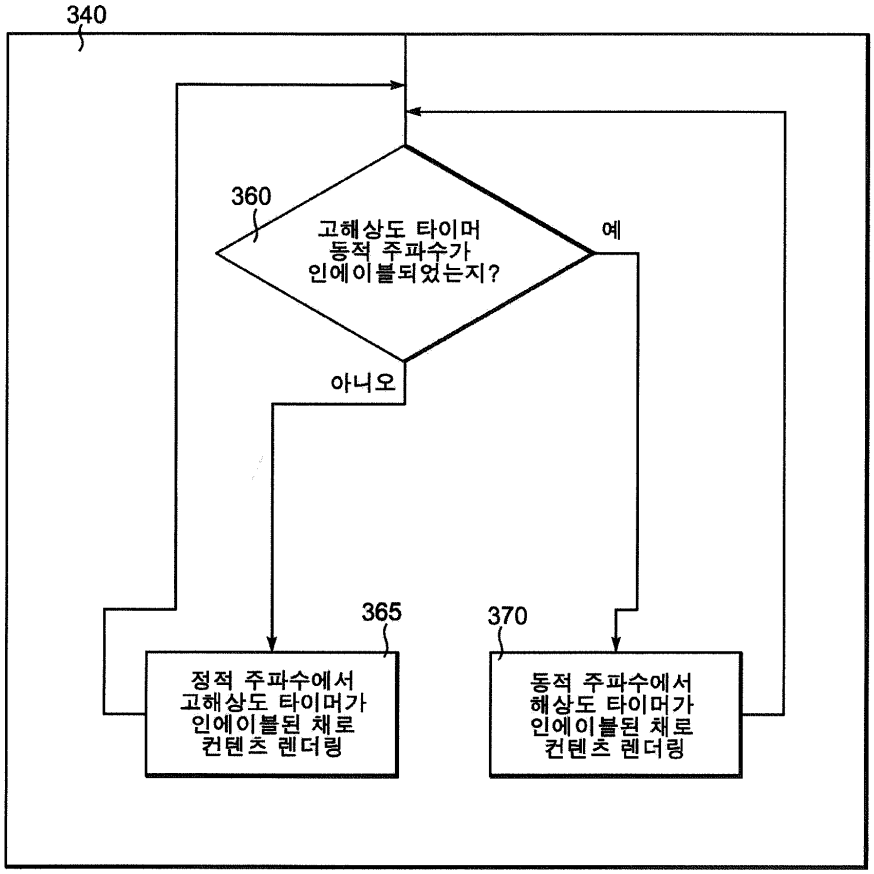
도면10


현재 디스플레이되는 버퍼 보고
현재 디스플레이되는 버퍼 보고
DWM 합성
하드웨어 큐
100
버퍼 m으로 렌더 (A)
285
287
275
280
290
DWM 프로세스
105
애플리케이션 생성 컨텐츠
110
(A)
(B)
계류중인 플립
115
(A)로 플립
온 스크린 디스플레이
120
버퍼 1
버퍼 2
리프레시 1
리프레시 2
리프레시 3
리프레시 4
버퍼 1
버퍼 2
버퍼 m

도면11


시작
300
305
전체화면
전용 모드?
아니오
320
대기시간 민감 컨텐츠 감지
310
고해상도 타이머를
디스에이블한 채로
전체화면 전용
모드로 렌더링
315
단계(305)로 돌아감
323
대기 시간
민감 컨텐츠?
아니오
예
340
고해상도 타이머를
인에이블한 채로
컨텐츠 렌더링
345
DWM 합성
325
고해상도 타이머를
디스에이블한 채로
윈도우 모드로
렌더링
350
동적 리프레시 및 버퍼 플립
330
단계(305)로 돌아감
355
고해상도 타이머가 버퍼가
이제 이용가능하다는 것을
합성자로 보고하게 함

도면12


340
360
고해상도 타이머
동적 주파수가
인에이블되었는지?
예
아니오
365
정적 주파수에서
고해상도 타이머가
인에이블된 채로
컨텐츠 렌더링
370
동적 주파수에서
해상도 타이머가
인에이블된 채로
컨텐츠 렌더링